VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Správa a konfigurace bezdrátové sítě založené na platformě Mikrotik

Administration and Configuration of Wireless Network Based on the Mikrotik Platform

2010 Martin Švidrnoch

Poděkování:

Chtěl bych poděkovat mému vedoucímu bakalářské práce panu Ing. Liboru Michálkovi, Ph.D. za jeho cenné rady při zpracovávání praktické i teoretické části mé bakalářské práce.

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: …….

Martin Švidrnoch

ABSTRAKT

Cílem mé práce je skriptování a jeho praktická implementace, která řeší problematiku rezervaci a řízení datových toků, bezpečnosti a zálohováni dat v malé síti, která je připojena k Internetu, za pomocí platformy Mikrotik . V teoretické části se seznámíte se analýzou a možnostmi zařízení, jež je ovládáno samostatným operačním systémem Mikrotik RouterOS. Praktická část nám ukáže funkční řízení datových toků, priorizaci datové služby VoIP, kontrola dostupnosti IP adres, která nám poskytne vhodné informace při řešení problému a jeho rychlejší odstranění a zálohy celého systému Mikrotik.

Klíčová slova: Mikrotik, RouterOS, Skriptování, QoS

ABSTRACT

The main point of my work is scripting and his practice implementation, which solves problem of reservation, control of data flows, security and backup in a small network, which is connected to the Internet thru the Mikrotik platform. In theoretical part you acquaint with analyses and options of device, which is control single operation system Mikrotik RouterOS. The applied part of bachelor work shows us functional control of data flows, prioritization data service called VoIP, checking availability IP addresses, which offer to us properly information in solving problems and faster removing and backup whole system Mikrotik.

Key Words: Mikrotik, Router0S, Scripting, QoS

Seznam použitých symbolů a zkratek

3G	3rd Generation - zkratka pro třetí generaci mobilních telefonů.
AP	Access Point - slouží jako přístupový bod pro připojení k Internetu.
CIR	Committed Information Rate - je garantovaná minimální průchodnost sítě.
DHCP	Dynamic Host Configuration Protocol - slouží pro automatické přidělování IP adres počítačům v síti.
DNS	Domain Name Systém - překládá doménové jména na IP adresy.
ESP	Encapsulating Security Payload - je členem protokolu IPsec.
FTP	File Transfer Protocol - slouží pro přenos souborů mezi počítači.
FUP	Fair User Policy - dodržuje, aby všichni uživatelé v síti měli stejně rychlé připojení.
GUI	Graphical User Interface - grafické rozhraní pro uživatele.
HTB	Hierarchical Token Bucket - jedná se o rychlejší náhradu pro třídu řízení front.
HTML	HyperText Markup Language- značkovací jazyk pro hypertext.
HTTP	Hypertext Transfer Protocol - internetový protokol, sloužící pro výměnu hypertextových dokumentů ve formátu HTML
ICMP	Internet Control Message Protocol - kontroluje zda je síť dostupná. Používá jej nástroj ping.
IDE/ATA	Integrated Drive Electronics/AT Attachment - je to počítačová sběrnice pro připojení zařízení k uchovávání dat.
IPsec	IP security - je to bezpečnostní rozšíření IP protokolu na sítově vrstvě.
LAN	Local Area Network - označení pro počítačovou síť.
MAC	Media Access Control - jedinečná adresa síťového zařízení.
MIR	Maximum Information Rate - maximální šířka pásma.
MK	Zkratka pro Mikrotik.
MPLS	Multiprotocol Label Switching - řídí data z jednoho uzlu na druhý.
NAT	Network Address Translation - překládá síťové adresy.
NTP	Network Time Procotol - protokol pro synchronizaci hodin z Internetu.
OS	Zkratka pro Operační systém.
PCQ	Per Connection Queuing - algoritmus pro řízení front.
PFIFO/BFIFO	Packet First In First Out/Bytes - druh front, které fungují na principu, první příchozí jde první pryč. Písmena P a B deklarují, zda se jedná o frontu s Pakety nebo Bajty.
QoS	Quality of Service - protokol pro řízení datových toků v síti.
QT	Queue Tree - stromová fronta.
RB	RouterBoard - hardware od společnosti Mikrotik.

RED	Random Early Detection - řídící algoritmus.
RTCP	RTP Control Protocol - doplňuje RTP protokol.
RTP	Real-time Transport Protocol - protokol pro doručování obrazových a zvukových dat.
SATA	Serial ATA - sériové sběrnice pro připojení počítačových zařízení.
SD/SSD	Secure Digital/Solid-state drive - paměťová media.
SDP	Session Description Protocol - popisuje vlastnosti relace multimediálního přenosu dat.
SFQ	Stochastic Fairness Queuing - řídící algoritmus.
SIP	Session Initiation Protocol - protokol pro přenos signalizace v Internetové telefonii.
SSH	Secure Shell - označení pro program i protokol v počítačové síti, který umožňuje komunikaci mezi počítači.
SSID	Service Set Identifier - identifikátor pro každou bezdrátovou počítačovou síť.
TCP/IP	Transmission Control Protocol/Internet Protocol - sada protokolů pro komunikaci přes Internet (Primární transportní protokol / Protokol síťové vrstvy IP).
UDP	User Datagram Protocol - je to protokol transportní vrstvy orientovaný na zprávy.
USB	Universal Seriál Bus - sériová sběrnice sloužící pro připojení periférií k počítači.
VoIP	Voice over Internet Protocol - je to technologie umožňující přenos digitálního hlasu přes Internet.
VPN	Virtual private network - zkratka pro virtuální privátní síť.
WDS	Wireless Distribution Systém - systém pro připojení k bezdrátové síti.
WEP,WPA	Wired Equivalent Privacy, Wi-Fi Protected Access - šifrovací metody pro zabezpečení bezdrátových sítí.
WLAN	Wireless LAN - standart pro lokální bezdrátové sítě.
WWAN	Wireless Wide Area Network - bezdrátová sít která pokrývá rozlehlé grafické území.
WWW	World Wide Web - je to celosvětová soustava propojených hypertextových dokumentů.

OBSAH:

1. Úvod

Celosvětový fenomén s názvem Internet je počítačová sít, která se rozšířila mezi všechny obory lidské činnosti a je běžným standardem ve světě informačních technologií. Jde o globální síť, kterou v dnešní době využívá více než miliarda uživatelů a poskytuje nesmírné množství služeb (například e-mail, telefonování, televize, rádio, www a spoustu dalších služeb, důležitých pro lidskou společnost). Internet je založen na komunikačních protokolech TCP/IP (primární transportní protokol/protokol síťové vrstvy) vzniklých v 80. letech 20. Století. V té době si nikdo neuměl představit, jak masivně se rozšíří mezi společnost a jak náročná bude práce správce. Jak složité budou konfigurace pro správný chod Internetu, jeho bezpečnost v lokálních sítích.[17]

V dnešní době, kdy na každém kroku máme spoustu sofistikovaných a inteligentních zařízení, není zas takový problém správné vytvoření sítě pro běžné uživatele. Řízení a správa této sítě již požaduje potřebné kvalifikace a k tomu nám slouží nejrůznější zařízeni. Máme zařízeni, cenově levnější, které nám slouží například pouze k připojení na Internet s nejjednodušším zabezpečením, až po zařízení, jež jsou svými vlastnostmi vyspělé natolik, že na nich můžete nastavit široké spektrum informací pro realizaci složitějších, bezpečnějších sítí. Jedním z těchto zařízení je Mikrotik RouterOS.

Zařízení Mikrotik RouterOS je router s operačním systémem na bázi Linuxu. Jde o jedno ze složitějších zařízení a je vhodný pro bezdrátové spojení. Hlavní výhodou tohoto zařízení je, že mají vlastní operační systém a že jsou osazeny různě silnými procesory od slabších, až po velmi silné tzv. "core routery". Pokud bych měl napsat o všech možnostech tohoto zařízení, tak bych strávil čtením nejeden den. Více o rozboru tohoto zařízení popíšu hned v první kapitole, kdy představím základní práci s tímto zařízením a popis některých funkcí. Pro zjednodušení správy těchto routerů se dají vytvořit skripty, které však vyžadují větší znalosti z oboru skriptování na platformě mikrotik, ale mají výhodu v implementaci na jednotlivých zařízeních.

V druhé kapitole podrobně popíšu skriptovací jazyk Mikrotiku, se kterým disponuje. Umožňuje základní práci se smyčkami, deklarování proměnných a vytváření podmínek. K vytváření samotných skriptů je potřeba lepší znalosti technické angličtiny. Skripty jsou uloženy přímo v daném zařízení. Jejich využití se dá použít ke zjednodušení spousty pravidel, které musíme ručně vkládat a psát. Ušetříme nesmírně moc času a v případě větších sítí, kdy se správce stará o více zařízení, opravdu ocení, když přijde a hned, jednoduchým nahráním, přidá skript do zařízení.

Od praktických kapitol, ve kterých se dovídáme jak vlastně Mikrotik RouterOS pracuje, a kde se setkáváme s problémy i výhodami skriptování, se dostáváme k praktické části kapitol bakalářské práce a to k naprogramování skriptů, jež budou řešit problémy dnešní doby (jako například „Quality of Service“, což je však široký pojem a tak se představí jen část této problematiky - telefonování přes Internet, FUP a kontrolu toku dat). Co se týče bezpečnosti Mikrotik je velice bezpečné zařízení, a proto skript, který by zajišťoval včasnou opravu testovaného zařízení, vede k vyřešení problému a ke zlepšení vztahů mezi uživatelem a poskytovatelem Internetu. I nejzkušenější správce se neobejde bez zálohy svého zařízení kvůli okolním vlivům, které mohou nastat (např. bouřka, znehodnocení zařízení). Skript, který bude zasílat zálohu na Internet, tak bude chránit zařízení před ztrátou dat. Sledování skriptů v reálných podmínkách a jeho testování bude uvedeno v poslední kapitole.

2. Analýza platformy Mikrotik a Routeros

2.1 Co je to Mikrotik

Mikrotik byl vytvořen v Litvě a původně byl znám jako Mikrotikls Ltd., ale jeho celosvětový název je Mikrotik. Byl vytvořen v roce 1995, začleněný v roce 1996. Jde o celosvětový projekt pro bezdrátové připojení pro poskytovatele Internetu. V tomtéž roce byl vyvinut vlastní software pro počítače intel, které měly funkci routeru. Jejich hlavním úmyslem bylo prodávat v nově vznikajících bezdrátových technologiích tato zařízení. V roce 2002 měla více než 20 zaměstnanců a v roce 2007 už více než 70 zaměstnanců od programátorů, administrativních pracovníků, technickou podporu, až po prodávající a další zaměstnance. Pro Mikrotik se vyvíjejí stále nové platformy a jsou pořádána pravidelná školení na toto téma. Mikrotik RouterBoard si můžeme prohlédnout na obrázku Obr 2.1 [20], [21], [1]

Obr 2.1.Mikrotik RouterBoard RB411

2.2 RouterOS

RouterOS je hlavním produktem firmy Mikrotik a je jednou z mnoha distribucí systému Linux Je znám jako Mikrotik RouterOS, tato verze Linuxu je ovšem plně komerční. RouterOS je tedy vlastní operační systém založen na Linuxu. Hlavním úkolem tohoto operačního systému je vybudování lokálních sítí pro poskytovatele, kteří vyžadují vyšší nároky a možnosti s konfigurací routeru. Údržba routeru není obtížná a zvládnou ji i méně pokročilí technici. RouterOS má mnoho funkcí. Mezi ty důležitější patří například firewall, routing, bezdrátový přístupový bod, „Quality of service“, VPN sítě, nebo skriptování. To je však jen výčet některých z funkcí, jelikož možnosti tohoto routeru jsou velice rozvinuté. Pro komunikace s routerem slouží graficko-uživatelské rozhraní-GUI, které se nazývá Winbox, jež však není jediným možným přístupem k Mikrotiku. Dá se k němu připojit přes Telnet nebo sériovou konzoli, která je vhodná zvláště při prvním nastavení. Tento operační systém má největší využití u bezdrátových spojů 802.11 a/b/g/n. RouterOS je univerzální systém a mé velkou podporu uživatelů z různého diskusního fóra, kde jsou nejrůznější příklady a návody na konfigurace. [3],[4]

2.3 Hardwarové požadavky

Mikrotik RouterOS se může provozovat buď jako běžný počítač, který však musí mít minimálně 100MHz a paměť 64 MB, doporučena je však konfigurace s vyšší konfigurací, a další x86 kompatibilní hardware. Dále se může využít speciální platformy, která byla vytvořena přímo pro tento systém. Nazýváme jej RouterBOARD znám pod zkratkou RB a typové číslo. Výhoda routerboardů je v jejich velikosti. Jsou umístěny v montážních krabicích, které jsou různě velké podle typu routerboardů. Každý typ má určitý počet ethernet portů, počet miniPCI slotů a především jsou různě výkonné, podle použitého procesoru. Například RouterBoard RB411 má 1x ethernet, 1x MiniPCI, konektor pro připojení sériové konzole a procesor s 300MHz.

Mikrotik RouterOS podporuje standardní IDE/ATA zařízeni ale také SATA, USB flash disků, SD karty a SDD disků. K nainstalování operačního systému je potřeba alespoň 64 MB volného místa. Podporuje také velké množství síťových rozhraní (jako jsou ethernet porty, bezdrátové karty pro různá pásma-802.11a/b/g/n u novějších verzí i dokonce 3G modem). [20]

2.4 Komunikace s Mikrotikem

Každý Router používá pro komunikaci nějaký prostředek, ať už je to webové rozhraní, nebo program pro správu. Mikrotik RouterOS má možností pro správu hned několik. Nejznámější a nejpoužívanější je Winbox**,** který je přímo pro toho zařízení vyvinut. Winbox je konzole, která používá pro konfiguraci a správu pomoc grafického uživatelského rozhraní. Funkce ve Winboxu jsou velice podobné funkcím přes konzoli. Do Winboxu se můžeme připojit přes http, který je standardně na TCP portu 80 nebo si jej můžu stáhnout ze stránek výrobce. Dalším prostředkem ke komunikaci je Telnet. Mikrotik RouterOS má v sobě vestavěný Telnet server i klienta. Hlavním cílem serveru je umožnit standardní metody propojeného terminálu mezi zařízeními. Je zpřístupněn přes port 23. Telnet klient se používá pro připojení k jinému hostu v sítí přes Telnet protokol. Obdobou Telnet protokolu je protokol SSH*(secure shell),* který pracuje na stejném principu, ale spouští se přes port 22, který se dá změnit. SSH je novější protokol, proto bylo jeho cílem nahradit protokol Telnet. Mikrotiky mají v sobě zabudovaný konektor, který slouží pro sériovou linku. Sériová linka musí být typu female-female. Tato sériová linka slouží pro komunikaci. V operačním systému Windows můžete například použít hyperterminal. Pro komunikaci je třeba nastavit rychlost komunikace na 115000 b/s. Řízení toku se nastaví na- žádná, ostatní parametry necháme defaultní. Základní menu prostředí Winbox máme na obrázku Obr 2.2.[7]

Obr2.2 Winbox menu

2.5 Funkce Mikrotiku

2.5.1 Bezdrátové sítě

V dnešní době si pomalu nedokážeme představit, že by nebyly bezdrátové sítě, tam kde je kabel nedosáhne nám jsou velkou oporou, ať už to jsou lokální bezdrátové sítě (WLAN) nebo dálkové sítě (WWAN) Mikrotik podporuje bezdrátové sítě s čipsety bezdrátových adaptérů a to Atheros a Prism, které pracují s normami 802.11a/b/g/n. Dokáží pracovat v režimech klient a přístupový bod je z angl. Access Point-AP. Bezdrátové spojení na mikrotiku má hodně vlastností a možností v nastavení.

Nstreme a Nstreme2 protokoly – Nstreme je sada přenosů, která se týká multipoint spojů tedy spojů point to point. Nstreme2 nám umožňuje použít 2 bezdrátové karty jak pro přijatá data, tak i pro odeslaná.

„**Client polling**“- je funkce, která snižuje přístupovou dobu k médiu tzn., že karta neověřuje vysílání dalších karet. Výhodou je protokol, který neomezuje tvořit spoje na velkou vzdálenost, aniž by docházelo k velkým poklesům v rychlosti na rozdíl od běžných bezdrátových spojení.

Další možnosti jako WEP, WPA, WPA2 šifrování pro bezpečnost, možnost vytvoření virtuálního přístupového bodu, bezdrátový distribuční systém WDS, a spoustu dalších funkcí.

Nastavení bezdrátového připojení v mikrotiku není problém a to díky přehlednému grafickému rozhraní winbox. Zde se může vše nastavit a to vše lze nastavit pomocí příkazů. Pro představu výpis bezdrátového připojení v modu pseudobridge.

```
[admin@vsbPracovni] /interface wireless> print
Flags: X - disabled, R - running
 0    R  name="wlan1"  mtu=1500  mac-address=00:0B:6B:DA:93:19
arp=enabled
      interface-type=Atheros  AR5213  mode=station-pseudobridge
ssid="maras"
      frequency=2437  band=2.4ghz-b  scan-list=default  antenna-
mode=txa-rxb
      wds-mode=dynamic    wds-default-bridge=wds-bridge    wds-
ignore-ssid=no
      default-authentication=yes        default-forwarding=yes
default-ap-tx-limit=0
      default-client-tx-limit=0     hide-ssid=no     security-
profile=default
      compression=no
```

Příkaz „print“ vypíše podrobné nastavení a informace o sekci, ve které se právě nacházíme. V našem případě tedy vypiš vše, co je v interface wireless. Name, je název rozhraní, mode je nastaven jako pseudobridge pro pásmo 2,4GHz a SSID je název bezdrátové sítě pro jejich rozlišení [21], [3], [22]

2.5.2 Firewall

Firewall je síťové zařízení, které slouží k zabezpečení a řízení komunikace mezi sítěmi z různou důvěryhodností. Příkladem může být firewall mezi Internetem představujícím zónu s velmi nízkou důvěryhodností a privátní sítí LAN s vysokou důvěryhodností. Firewall provádí filtrování paketů, a tím mají bezpečnostní funkci a používají se pro správu dat, které jdou z routeru do routeru a skrz něj. K zabránění neoprávněnému přístupu slouží překlad adres, které jsou přímo připojené k síti na routeru. Mikrotik se dá použít tedy jako bezpečný hardwarový firewall. Firewally mají mnoho funkcí. Některé z nich si teď popíšeme:

- **Filtrování paketů**- Odmítání paketů od neautorizovaných uživatelů a pokusy o připojení k neautorizovaným službám.
- **Překládání síťových adres** (NAT)- Překládá IP adresy hostů v interní síti, které jsou odstíněné a skryté před monitorováním zvenčí. Mikrotik využívá Source NAT – překlad zdrojových adres a Destination NAT – překlad cílových adres.
- **Šifrovaná autentizace**- Povoluje vstup do interní sítě na základě autentizace uživatelů externích sítí
- **Filtrování obsahu**- Filtrování obsahu umožňuje blokování přístupu interních uživatelů k určitému obsahu na Internetu. Například rasismus a pornografie.

Firewall poskytuje funkce pro využití interního připojení, směřování a značení paketů pomocí mangle, ten nabízí široké využití v rámci celého routeru, nejčastěji pro seskupování paketů při kontrole toku dat.

Další významnou roli hraje u firewallu filtrování. Filtry dělíme na tři řetězce: *vstupní* neboli input, který zahrnuje pouze pakety určené pro daný router, *výstupní* tedy output, kde pakety jsou generované přímo routerem a *forvard,* kdy pakety routerem pouze procházejí. U tohoto řetězce máme více druhů provozu.

Bridgeovaný provoz - mód, který nezatěžuje provoz samotného routeru a jeho funkce je rychlé předávání paketů. Kontroluje MAC adresy směřovaného paketu a podle toho se rozhoduje, zda předá paket dál. V IP provozu, prochází nejdříve prerouting řetězem, vstupuje do forward, a vystupuje postrouting řetězem.

Routovaný provoz – má tři typy vyhodnocování a postupu daných paketů Provoz směrovaný pro samotný router vstoupí přes prerouting do input řetězu a je zpracován vnitřně. Provoz generovaný routerem odchází output řetězem do postrouting, kde může být dále zpracován a přizpůsoben našim požadavkům. Provoz pouze prostupující. Nejprve projde prerouting řetězcem, kde může být náležitě upraven, dále forward-em a vystoupí přes postrouting, kde můžeme opět uplatnit různá pravidla pro zpracování. [2]

2.5.3 Směřovaní

Směřování sítí je hlavním posláním Mikrotiku RouterOS. Směřování slouží k určování cest paketů v sítích TCP/IP. V systému mikrotik máme dva základní druhy směřování paketů- statické a dynamické. Statické směřování přidává správce sítě k směřování provozu do vzdálených sítí a také pro specifikování „gataway“ neboli výchozích bran v dané síti. V uživatelském rozhraní Winbox se zadávají statické routery v menu *IP->routes*. Dynamické směřování nám automaticky přidá routery po přiřazení IP adresy k adaptéru. Lze také využít speciálních protokolů pro dynamické směřování.

2.5.4 Fronty

Fronty v Mikrotiku jsou používány k omezení a priorizaci provozu. Mohou být použity pro omezení pro dané IP adresy, protokoly, porty a dalších parametrů. Pro priorizaci paketů, nastavení provozu pro rychlejší webové vyhledávání, aplikování limitů například v noci nebo ve dne nebo podle zadaného času a mnoho dalších využití. Existují dva rozdílné typy jak nakonfigurovat fronty v Mikrotiku RouteruOS. Pomocí Queue Simple nebo Queue Tree. Mikrotik také podporuje základní sadu front, jako jsou PFIFO/BFIFO, SFQ, RED, PCQ a implementace v Mikrotiku RouterOS je založena na „Hierarchial Token Bucket“ (HTB), které dovolují vytvořit hierarchické struktury fronty a určit vztahy mezi frontami.

Queue simple- její navržení vyplývá již z názvu, je to jednodušší verze řazení pro každodenní úkoly jako jsou například klientské limity download/upload dat atd. Dnes se již moc nepoužívá.

Queue tree- Byl sestrojen pro složitější úkoly jako jsou priorizaci globální politiky, limitování skupin uživatelů, které ovšem vyžadují označení paketů v /ip firewall mangle, tudíž je to již sofistikovanější řešení, v dnešní době to více používané.

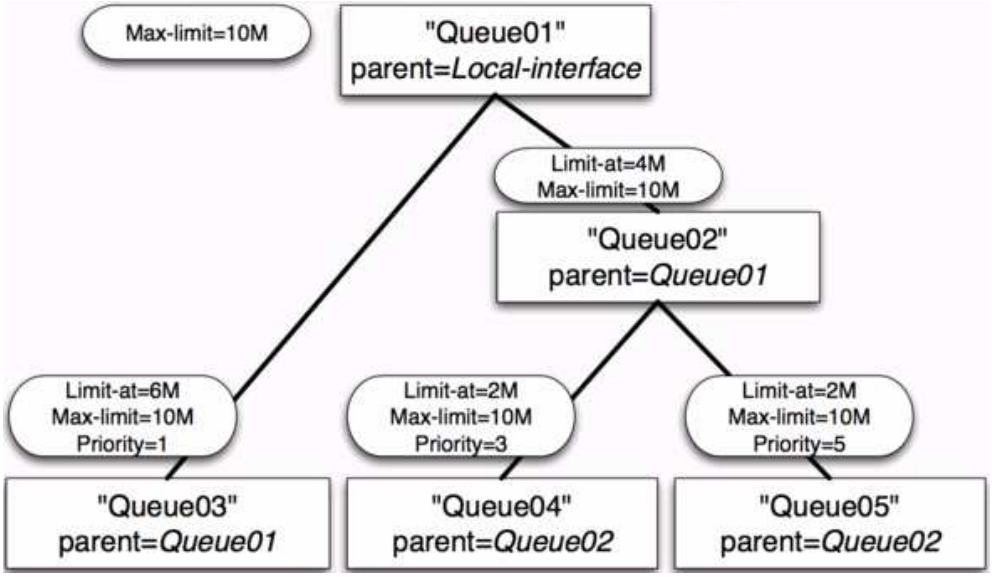
PFIFO/BFIFO - jsou to fronty, u nichž první písmeno znázorňuje, jestli bude fronta řazena pomocí paketů u PFIFO a nebo pomocí bajtů u BFIFO. Zkratka FIFO znázorňuje zda-li pakety nebo bajty, které přišly do fronty jako první, půjdou jako prví dál z angl. First-In First-Out. Tyto fronty nemají moc využití pro omezení rychlosti, jelikož jediným parametrem je jejich velikost. Fronty při nabytí této velikosti jsou další pakety/bajty zahozeny.

SFQ - Stochastic Fairness Queuing - Algoritmus SFQ propouští pakety podle nastavené velikosti zásobníku pro jednotlivé části tak, aby nedocházelo k potlačení datových toků a každé spojení bylo obslouženo. SFQ jak již má v názvu, se snaží o to, aby bylo množství dat spravedlivě rozděleno v daném provozu a také není vhodný pro omezení rychlosti.

RED - Random Early Detection - je řídící mechanismus který zabraňuje přetížení sítě pomocí náhodně zahozených paketů Ve chvíli kdy se naplnění fronty začne přibližovat velikosti parametru *red-min-treshold*, systém začne lineárně zahazovat náhodné pakety až do té doby než dojde k naplnění velikosti fronty a dosáhneme parametru *red-max-treshold*, v této chvíli se zahazují veškeré pakety, až do doby než se fronta alespoň částečně neuvolní. Vlastnosti TCP protokolu nám zpomalí rychlost odesílaných paketů. UDP(User Datagram Protocol*)* tuto vlastnost nemá tudíž RED nemá vliv na přizpůsobení rychlosti, nedochází však ke zhoršení kvality linky kvůli zahazovaným paketům.

PCQ - Per Connection Queuing – Tento alogorytmus je podobný SFQ ale na rozdíle od něj dokaže nahradit jeho drobné nedostatky. PCQ si vytváři podfronty řízené parametrem pcq-classifier a ty se klasifikují z cílové a zdrojové adresy nebo portu. Jde o velice jednoduchý algoritmus, který je praktický a proto je oblíbený.

HTB - Hierarchical Token Bucket- je to nejpoužívanější nástroj pro řízení datových toků. Povoluje vytvořit hierarchickou strukturu front a určit vztahy mezi frontami jako „parent-child“ nebo „-child“. Můžeme tedy vytvářet velice propracovanou stromovou strukturu. Jakmile má fronta alespoň jednoho potomka, stává se automaticky vnitřní frontou. Všechny fronty, které nemají potomkam, jsou listové fronty. V Mikrotiku RouterOS je důležité specifikovat možnosti rodiče, aby bylo možné přidělit frontu jako potomka do další fronty. Každá fronta v HTB má dvě měřítka limitů. Parametr *limit-at* je rychlostí, která je zaručená -CIR (Committed Information Rate) a *max-limit* je maximální možnou dosažitelnou rychlostí MIR (Maximum Information Rate). Dalšími možnostmi v nastavení priority, která je zodpovědná za distribuci zbývajícího provozu rodičovské fronty na frontu potomka, tak aby byl schopný dosáhnout max-limitu. Pro krátkodobé navýšení aktuální funkce nám slouží funkce Burst, od které se odvíjí funkce jako Burst-time-pro výpočet času průměrné aktuální rychlosti, Burst-treshold-je hodnota průměrné rychlosti a Burst-limit –která souvisí s Burst-treshold, do které se uplatní navýšení rychlosti. Příklad HTB uvádím na obrázku Obr 2.3. [11],[16],[1]



Obr 2.3. Příklad rozdělení HTB do 5 front.

2.5.5 QoS

Chceme-li v síti provozovat multimediální služby jako jsou telefonováni přes Internet, sledovaní televize přes Internet, video konferenční hovory nebo klasické aplikace se specifickými a náročnějšími požadavky, musíme k tomu sít přizpůsobit k tomu nám pomáhá právě Quality of Service. QoS je tedy souborem technik, které dokáží řídit ztrácení paketů, šířku pásma, zpoždění nebo fázové chvění a rozdělují se na určité úrovně.

- Best-effort-klasické připojení bez garancí
- Differentiated-zde jsou třídy rozděleny podle požadavků
- Guaranted-požaduje přidělení určité množství síťových zdrojů

Můžeme tedy říct, že router může upřednostnit a dát konečnou podobu síťovému provozu. QoS například využívá HTB nebo PCQ, které jsme si představili již v předchozí kapitole. Více si o QoS řekneme v praktické části. [9]

2.5.6 Ostatní služby

Mikrotik je velice rozsáhlý a výčet všech jeho funkcí. Měl by mít veliký rozsah. Proto ani v této kapitole nebudou všechny služby v Mikrotiku RouterOS, ale pouze ty které jsou v praxi více vyhledávané a jsou proto důležitější.

- **Hotspot** - Mikrotik Hotspot brána se umožňuje připojit k veřejné, sítí pro klienty využívající především bezdrátový sítě. Uživateli se předloží přihlašovací obrazovka, jakmile splní požadavky pro přihlášení bude mu umožněn přístup na Internet.
- **VPN** – Slouží pro bezpečné připojení skrz otevřené sítě Internetu nebo pro připojení do vzdálené sítě kde je využito šifrované spojení. Mikrotik RouterOS podporuje různé metody a protokoly. Jedním z nejznámějších je Ipsec-tunel, bod-bod tunel, ESP protokol a další. Využití se nalezne především připojením do bankovních sítí, kde je vyžadována veliká ochrana.
- **MPLS**- *Multiprotocol Label Switching*- Může být použita jako náhrada pro IP routerování. V síti MPLS, jsou přiřazeny datovým paketům štítky a rozhodnutí se přijímá výhradně nad obsahem tohoto označení, aniž by bylo třeba koumat paket sám o sobě.

- **FTP server**- Mikrotik RouterOS má v v sobě implementovaný FTP server. Je především určen pro softwarové balíky, které slouží k aktualizaci nebo potřebná rozšíření. Nahrávání je totožné s konfiguračním skriptem. RouterBoard většinou nemá dost velké paměti, což je docela velká nevýhoda. Většinou to stačí pro softwarová balíky. Pro komunikace využívá porty 20 a 21.
- **DHCP klient a server**- je protokol pro konfiguraci dynamických hostů a umožňuje snadné rozdělení IP adres v sítí Mikrotik RouterOS podporuje oba módy jak klienta, tak server. Tento protokol je v podstatě nezabezpečený, takže by měl být používaný pouze v zabezpečených sítích. DHCP může být použit i také s funkcí Hotspot, která bude sloužit jako ověřování pro DHCP klienty.
- **DNS**-Router Mikrotik dokáže nastavit systém do režimu DNS cache, může být nastaven jako primární DNS server a komunikuje s klienty přes UDP a TCP na portu 53. Pomocí vnitřní cache paměti dokáže minimalizovat požadavky posílaných externím DNS serverům.
- **NTP-** Slouží k synchronizaci lokálních hodin pomocí vzdáleného serveru. Je to velice užitečná pomůcka udržuje totiž správný čas, který využijeme například při funkci „Schedule“, kde je nastaveno např. kdy se mají spouštět skripty atd.
- **ICMP bandwith**- slouží pro monitorování propustnosti linek. Spouštíme jej příkazem */tool ping-speed*

Systém Mikrotik se snaží být univerzální pomůckou pro každého uživatele, jelikož každý uživatel má rozlišné potřeby v použití tohoto zařízení. Rozšiřování a technická úroveň v elektronice jde neustále kupředu a proto se vývoj softwaru se programuje dle požadavků provozovatelů. Zda Mikrotik RouterOS obstojí v rychlém vývoji sítí,se může jen spekulovat, ale zatím má ty nejlepší předpoklady. Jeho využití má velmi rozšířené možnosti za rozumnou cenu. [3], [1]

3.Skriptovací jazyk pro Mikrotik, syntaxe a využití.

3.1 Skriptování v Mikrotiku

Skriptovací jazyk je programovací jazyk, který nám umožňuje spravovat jednu nebo více softwarových aplikací. Využití skriptování v Mikrotiku umožňuje použít jeho vlastní programovací jazyk k ulehčení a použití funkcí v zařízení. Díky skriptování můžeme využít příkazů, které jsou vykonané v příkazové konzoli. Skripty mohou být uloženy v routeru, nebo mohou být vloženy přímo do konzole. Skripty jsou uloženy v menu „/system skript“. Každý skript má vlastní název a mohou být spouštěny ve specifickém času nebo v daných časových intervalech. Může být také spouštěn na základě nějaké události jako je například nástroj „netwatch“, který zkoumá, jestli adresa reaguje na odezvu. Skripty mohou být spouštěny i pomocí jiných nástrojů jako „traffic monitoring“ a „systém schelduler“ Právě pomocí plánovače jde nastavit, jakým způsobem mají být skripty spouštěny. Jednoduchým příkladem je například: `add interval=1d name="zaloha_na_mail" on-event=mail,` kde interval znamená, že se skript bude spouštěn každý den, z názvu zaloha_na_mail na událost email. Skripty mohou být také spouštěny jinými skripty. Samotný proces skriptování je velice náročný, jelikož tento programovací jazyk neprovádí žádný „debugging“ tedy při překladu programu se neukáží chyby. Ve vyšších verzích mikrotiku nastávají drobné změny, které mohou vést k nekompatibilitě skriptů. V Mikrotiku routeruOS je skript rozdělen na počet příkazových řádků a tyto řádky jsou prováděny jedna po druhé do té doby, než skript není kompletní nebo do té doby, než nastane nějaká chyba. Pomocí skriptů můžeme nastavovat nebo provádět různé kontroly routeru a nastavování nejrůznějších parametrů o jeho využití si povíme později. [5] ,[23]

3.2 Struktura skriptovacího jazyku

Souhrnný popis skriptování v Mikrotiku se skládá z názvu, zdroje, vlastníka, počtu spuštění, a kdy byl naposledy spuštěn. Název slouží pro odkazování, pro nástroje jako bylo například uvedeno v předchozí kapitole u plánovače. Není-li název vytvořen automaticky, popíše se jako „skriptN“, kde platí, že N je jednoznačné číslo, které se neustále zvětšuje. Zdrojem je myšleno vlastní jádro skriptu. Vlastník je název pro autora skriptu. Počet spuštění, má funkci počítadla, které se dá vynulovat příkazem „run-counter=0“ Poslední položka definuje, kdy byl skript naposledy spuštěn a také se dá vynulovat.

V Mikrotiku RouteruOS se používá následující syntaxe příkazu.

- **[prefix]**- Má dva typy „:“ nebo „/“ které určují, zda bude příkaz cesta nebo vnitřní výraz konzole.
- **[cesta]**- Je to relativní cesta k různé úrovni v menu. Nemusí být specifikována.
- **Příkaz**- Zde se vykoná příkaz některý z příkazů, které jsou k dispozici na nějaké úrovni v menu.
- **[uparametr]**- Nepojmenovaný parametr musí být specifikovaný, jestli jej příkaz vyžaduje.
- **[parametry]**- Je pořadí označených parametrů, které jsou následovány přeslušnými hodnotami.

Konce příkazů jsou ukončeny znakem „;“ nebo novým řádkem, ale někdy nejsou zapotřebí. Když jsou příkazy uvnitř jakých kolik závorek jako například (), [ ] nebo {} nepotřebují zakončení. Příkazy, které jsou uvnitř jiných příkazů, na jednom řádku začínají a jsou i ukončeny hranatými závorky. Tento proces se nazývá příkazová koncentrace.

Programy pro svou přehlednost využívají ve své zdrojovém kódu funkci poznámek nebo komentářů. V mikrotiku se tyto komentáře vytvářejí pomocí znaku „#“ .Tento symbol musí začínat na první pozici v kódu, nesmí byt před ním žádný symbol.

Jak již bylo řečeno dříve, skriptování nemá žádný překladač, tudíž je nutné dodržovat správnost psaní skriptovacího kódu. Důležité je dodržovat správnost mezer. Mezery nepoužívají mezi parametry *„from=“, „to=“ , „step=“ , „in=“ „do=“ , „else=“*. Dále si předvedeme příklad, kde se používají mezery a kdy ne.

```
{
:local a true;
:local b false;
# zde se mezery nepoužívají
:put (a&&b);
# zde se mezery vyžadují
:put (a and b);
}
```

Skript se rozděluje na tzv. oblasti. Jeho využití je například v deklaraci proměnných, které mohou být například využity jen pro danou oblast. Tyto proměnné nazýváme lokální a proměnné pro celý skript uvádíme jako globální. Proto můžeme říct, že jsou oblasti lokální a globální. Globální oblast je vytvářena automaticky a můžeme ji nazývat kořenovou oblastí. Tato oblast nemůže být zrušena. Pro definici svých vlastních skupin se využívají složené závorky „{ }”, které mohou omezit přístup jiným proměnným.

3.3 Datové typy

Datové typy definují v programování druh proměnných. Skoro každý programovací jazyk má některé datové typy již předdefinované. Mikrotik RouterOS používá následující datové typy.

- **number**-64bitové číslo které může být i hexadecimální.
- **boolean-**hodnoty „true“ nebo „false“.
- **string-** tvoří řetězce.
- **IP-** IP adresa.
- **internal ID-** hexadecimální hodnoty začínají znakem a znamenají, že každá položka v menu má přiděleno ID číslo.
- **time-** hodnoty data nebo času.
- **array-** hodnoty, které jsou uspořádány v poli.
- **nil-** defaultní proměnná.

3.4 Operátory

Mikrotik RouterOS umožňuje použití aritmetických operátorů pro jednoduché výpočty s čísly, IP adresami, časovými hodnotami řetězci a seznamy. Pro získání výsledků pomocí operátorů se musí uzavřít do závorek a výsledek bude návratová hodnota pro závorky. V mikrotiku RouteruOS se používají mnoho takových operátorů, a dělí se na několik skupin. Jako jsou aritmetické operátory, relační operátory, logické operátory, bitové operátory a operátory, které slouží k sřetězení, takže je můžeme nazvat slučovací.

- **Aritmetické-** `+ , - , * , _ , / ;`
- **Relační -** `<, > , =, <=, >=, !=;`
- **Logické-** `(!, not), (&&, and), (||, or), in`
- **Bitové-** `~, |, ^, &, <<, >>`
- **Slučovací-** `„tečka, čárka“`

Operátory, které nespadají do žádné z uvedených skupin (operátory jako [], (), ->, $), jsou využívány velmi často, proto je detailněji popíši. ***[ ]***- může obsahovat pouze jeden příkaz ***( )***- slouží jako podvýraz nebo seskupovací operátor. ***$***- nahrazovací operátor.

3.5 Příkazy

V Práci na počítači je směřování na program nebo skript prováděno pomocí příkazů a jejich funkce je provedení specifického úkolu.Mikrotik RouterOS má spousty příkazů, které nejsou závislé na aktuální úrovni v menu. Příkazy sice nemění konfiguraci přímo, ale jsou užitečné pro různé úkoly údržby.

3.5.1 Globální příkazy

- **beep -** Klasický bzučák pro vytváření signálu. Parametry pro jsou délka, kde se určuje jak dlouho signál trvat, a frekvence na které bude pracovat.
- **:delay** - Klasické zpoždění.
- **:environment print** - Vypíše informace o inicializovaných proměnných.
- **:find** - Je to vyhledávací příkaz, který hledá podřetězce uvnitř jiného řetězce, nebo nějakou hodnotu či prvek z pole v závislosti na typu argumentu vrací pozici, na které byla hodnota nalezena.
- **:put** - Přidává argumenty do konzole.
- **:len** - Je to řetězec, který vrací jeho délku nebo pole v závislosti na typu argumentu.
- **:typeof** - Tento příkaz vrací datový typ proměnné.
- **:pick** - Vrací rozsah prvků nebo nějaký podřetězec v závislosti na vstupní hodnotu a jestliže tato hodnota není zadána, vrátí všechny hodnoty z dané pozice až do konce řetězce nebo pole.
- **:log -** Vypíše zprávu do systémových záznamů. Využíván může být například pro výpis varování, informace, chyby atd.
- **:time -** slouží jako funkce času potřebného k vykonání potřebného příkazu.
- **:set -** Přiřadí novou hodnotu pro dané proměnné.
- **:to“<datový typ>“-** převádí poměné do zvoleného datového typu, například pro převedení do datového typu boolean bude vypadat příkaz *:tobool* <*proměnná*> Kromě datových typů může být také převeden proměnné do pole.

3.5.2 Cyklové příkazy

- **do** - Pracuje stejně jako ve většině programovacích jazyků. Bude provádět příkazy, dokud nebude splněná daná podmínka. Může být proveden v kombinaci s „while“ parametrem. Kdy tento parametr se provede až po vykonání příkazu a v případě, že výsledek bude „true“, tak bude příkaz provádět tak dlouho, dokud nenastane hodnota „false“. V kombinaci s „if“ parametrem znamená, že vykoná příkaz je-li splněna nějaká podmínka.
- **for** -Provádí příkazy pro daný počet iterací. Využívá parametry „from“ a „to“.
- **foreach** - Podobný příkazu for s rozdílem, že příkaz je prováděn pro každý prvek v listu. Využívá parametry „in“ a „do“.

3.5.3 Obecné příkazy

- **add** - Klasické přidání nového prvku.
- **remove** - Odstraní označený prvek.
- **enable** - Povolení označeného prvku.
- **disable** - Opak příkazu enable.
- **get** - Získá z vybrané položky hodnotu parametru.
- **export** - Odešle konfiguraci z aktuální položky v menu a jeho podskupiny. Většinou je zapisován s příponou .rsc . Příkazy mohou být navráceny import příkazem.
- **edit** - Slouží k úpravě položek.
- **print** - Slouží k výpisu prvků v menu. Tento příkaz je velice používaný při nastavování mikrotiku, abych se mohli podívat, zda je vše v pořádku. Vypíši si jej právě pomocí tohoto příkazu. Tento příkaz používá spoustu parametrů jako například: „from, detail, where, as-value, file, a mnoho dalších.

3.6 Využití

Samotnému využití skriptování v Mikrotiku RouteruOS se meze nekladou. Jeho využití je velmi rozsáhlé, ale můžeme jej pro některé funkce použít přednostněji. Skripty se píší, aby ulehčily práci. V mikrotiku jde vše nastavit ručně, proto jeho využití je hlavně otázkou komfortu za určitou cenu, a tou je znalost příkazů, skriptovací znalosti a Mikrotiku routeruOS samotného. Například skripty jako záloha nám zajišťuje určité bezpečí, když dojde k hardwarovému poškození zařízení. Nejčastější využití skriptování je ve FUP (*fair use policy)* tzn. skript pro omezení uživatelů, aby měli stejnou rychlost nebo skript pro přidávání nových uživatelů až po signalizace, které nás upozorňují na nejrůznější věci. Jako je např. základní oznámení při spuštění Mikrotiku.

Další možnosti jsou v oblasti QoS, bezpečnosti na Internetu, posílání různých důležitých oznámení a na telefon, kontrolu sítí a mnoho dalších. Každý provozovatel si pro psaní vybírá odlišné téma a záleží také na tom, jak je poskytovatel rozšířený. Skriptem totiž může nastavit zařízení jako dynamické DNS, IP adresy atd., a proto když si poskytovatel napíše jednou tento skript, nemusí jej nastavovat na mnoha Mikroticích znova a znova. Podstatnou nevýhodou skriptování je, že při vývoji nových verzí Mikrotiku RouterOS nemusí být kompatibilní se staršími verzemi a dochází tím k složitému přepracování (například verze 2.9.x měla problémy s globálními proměnnými při přechodu na verzi 3.x mezi apod.).

Já osobně si myslím, že když je skript dobře použit a není zbytečně složitý, je vhodné jej používat ve velké míře. Ve složitějších případech vzhledem k jeho složitosti na vytvoření bych je nedoporučoval. [5] ,[23]

4.Skripty řešící řízení QoS, bezpečnosti a zálohování na platformě Mikrotik.

4.1 Skript na řízení QoS s preferováním VoIP

Většina poskytovatelů se snaží, aby jejich zákazníci byli spokojení. K tomu jim napomáhá služba pro řízení datových toků známa jako „QoS“ z anglického „*Quality of service*“ můžeme tak řídit data tak, abychom tyto zákazníky uspokojili. Řídí tedy, aby nedocházelo k zahlcení sítě s následkem snížení kvality síťových služeb. Cituji[1]: „*Pomocí QoS se může např. nastavit maximální nebo minimální přenosové pásmo pro určitá data, prohlásit provoz za prioritní před ostatními nebo rozdělit provoz do kategorií podle nastavených parametrů. QoS se tedy snaží poskytovat uživatelům služby s předem garantovanou kvalitou, aby nedocházelo ke zpoždění, ztrátovosti nebo plýtvání šířkou pásma*.“

V mém skriptu jsem se zaměřil na vytvoření usměrňování toků, priorizaci důležitých paketů a portů které vyžadují více přenosového pásma, jako je například telefonování přes Internet. S omezením ostatních uživatelů při hovoru. Při telefonování přes Internet se využívají především dva protokoly SIP a RTP. SIP protokol slouží pro přenos signalizace. Jeho funkce je, že nejprve vyhledá účastníka, a zjistí jestli je dostupný pro telefonování. Jako je např. nedostupnost - obsazeno jiným volajícím. Poté zjišťuje u účastníka, jaký používá kodek, přenosovou rychlost atd. Následuje navázání spojení přes protokol SDP, který popisuje možnosti relace pro přenos dat, a odkazuje na RTP protokol. RTP protokol pak slouží pro přenos zvukových i obrazových dat. Slouží tudíž jako takový základ pro VoIP službu. Protokol je doplňován RTCP protokolem. Jejich rozdíly jsou, že RTP protokol odesílá pakety po milisekundách a RTCP v řádech sekund.

Nejprve nastavíme parametry „limit, speed, parent“. Limit parametr definuje pří kolika bytech dojde k omezení. Parametr speed udává rychlost v bytech na kolik bude omezovat. Parent bude definovat na jaké které položky v Queue tree se bude omezení vztahovat. Dále nesledují pracovní proměnné, které popíšu později.

Po deklarování proměnných je prováděna kontrola, aby nedocházelo k opětovnému spouštění již deklarovaných mangle paketů v „*/ip firewall mangle*“. Když se zamezí pravidlo pro opětovné spouštění, provede se příkaz, který provede vytvoření mangle pravidel, kde byl kladen velký důraz na vytvoření všech pravidel pro službu VoIP. Porty byly vyhledány z Internetu. ze seznamu TCP a UDP portů.

[1] [24]

Když jsou deklarovány všechna pravidla, tak nastavíme fronty do „Queue tree“ . Aby tyto fronty fungovaly správně, musí být dobře nastaveny mangle pravidla. Na základě mangle položek se totiž provedou fronty v „Queue tree“. Jediná fronta, které jsem věnoval větší pozornost je fronta pro VoIP. Kde jsou proměnné „*unikatnijmeno*“, která nám definuje jméno ve frontě a proměnná „*priorita*“ definující prioritu. Potom se provede příkaz pro provedení fronty na základě unikátního jména „voip“. Následuje omezení uživatelů podle stažených dat. Zde si připomeneme pracovní proměnné ze začátku skriptu. Jako jsou „*name, par, traf a max*“ U „name“ deklarujeme název v pro jednotlivé fronty v QT. U proměnné „par“ nastavíme pro jaký „*parent*“ se omezení bude vztahovat. Parametr „*traf*“ získá, kolik bytů bylo staženo. U proměnné „max“ máme získání maximálního limitu. Po této deklaraci máme přikaz, který porovná jestli hodnoty pro jednotlivé „*parenty*“ a jestli je počet stažených bytů vetší než proměnná „limit“ a zároveň se proměnná „max“ nerovná proměnné „speed“ ,tak se nastaví omezení daného QT. Tato funkce slouží jako ochranné FUP tzv. „Fair User Policy“, který se stará, aby všichni uživatelé měli stejné možnosti pro připojení k Internetu.

Ve skriptu je vytvořen schleduler, který každý týden restartuje skript, aby se vynulovaly počítadla, abych neomezil uživatele. Kdybych jsem nechal skript bez nástroje Schedule, tak by uživatel při nabytí maximálního limitu zůstal omezen. Proto mu garantuji mnou zvolené omezení na týden. [18], [19], [15],[13],[14]

4.2Bezpečnostní skript

Skript má funkci podobnou funkci jako různé programy s podporou řízení kontroly jako například Netwatch pro RouterOS, ale je vylepšen a je o různé podpůrné prostředky.

Nástroj Netwatch zaznamenává výpadky ping paketů na určitou adresu. Proto se využívá pro monitorování výpadků linek. V netwatch nástroji nastavujeme Host adresu, kterou budeme kontrolovat, interval, kde nastavujeme dobu mezi výpadky ping paketů, po které se vykonají potřebné příkazy. V záložkách UP/DOWN nastavuje příkazy. V případě DOWN se vykonávají příkazy, když host přestane vysílat. U UP příkazu je to přesný opak, tedy spouští se příkazy když host opět začne vysílat.

Můj skript pro bezpečnost se skládá ze dnou skriptů –IPbuilder a reportIPcheck. Ve zkratce, je ve skriptu IPbuilder vytvoření jednotlivých kontrolovaných IP adres a, reportIPcheck slouží pro zasílaní reportů na zadanou adresu. Ve skriptech je implementovaný schleduler potřebný pro automatické spouštění a kontrolu skriptů. Jediné tedy co je potřebné nastavit v address listu kontrolované adresy které musí jmenovat „IPcheck“ a vyplnit u něj komentář například, že adresa naleží nějakému routeru. Tento komentář se pak bude zobrazovat jak v odeslaném reportu, tak bude provázen všude v nastavení. [6],[12]

4.2.1 Skript IPbuilder

Ve skriptu nejprve deklarujeme proměnné „coment, counter, initialscript, ip “a jedna globální proměnná „IPcheckname “ Proměnná coment slouží pro deklaraci komentářů. Counter slouží jako počítadlo tj. když přidáme nějakou novou adresu do address listu tak nám jí počítadlo označí podle počtu kontrolovaných adres. Například když máme v address listu dvě adresy tak nám ji counter označí jako IPcheck1, jelikož jeho počáteční hodnota je nastavena na 0. Pro spuštění kontrolovaných adres z address listu je potřeba nejprve spustit počáteční skript, o který se stará proměnná initialscript. Poslední lokální proměnná v tomto skriptu je IP kde se nastavuje IP adresu z address listu. Globální proměnná nastavuje IPcheckname.

Dále nastavujeme, aby při opětovnému spuštění skriptu nedocházelo k spuštění dvou stejných skriptů, proto nastavíme aby se při spuštění skriptů pro kontrolování adresy se skripty které slouží pro kontrolu adres „IPcheck“ nejprve vymazaly. Toto nastavení provedeme totožně i u nástroje schleduler.

V dalším kroku nastavujeme pro kontrolované adresy označené jako „IPcheck“ parametry *IP*, *coment* a *initialscript*. Pro všechny ostatní adresy se provede tzv. „disable“ tedy deaktivovat je, aby mohl být provedeno nastavení initialscript.

Když máme provedené předchozí nastavení, které sice pořád spadá do nastavování IPcheck nastavení, dostaneme se k vytvoření tzv. podskriptu tedy skriptu který vytvoří další skript. Nejprve tedy nastavíme jméno skriptu s počítadlem-counter a nastavení politiky. Poté nastavuje zdroj celého podskriptu. Vygenerování skriptu je děláno předně pomocí značek \r\n které fungují jako klávesa „enter“, formálně tedy „newline“ s posunutím kurzoru na začátek řádku. Na začátku zdroje je vygenerován komentář o jakou kontrolovanou adresu se jedná a její IP adresa. Poté deklarování globálních proměnných IPcheck+(counter), IPDown a IPUp.

Následuje jádro celého podskriptu, kde je příkaz, ve kterém se provádí ping na adresu a pokud se 5krát neprovede změní se ze stavu „0“, který znamená, že IP pracuje v pořádku, na stav „1“ tedy na stav, který je ve stavu čekání kdy IP nepracuje správně. Pokud při opětovné kontrole, kdy už máme IPcheck nastaven na hodnotu „1“ a naše IP adresa stále nepracuje je hodnota nastavena na „2“. Při této hodnotě je nastavena proměnná IPDown s komentářem pro danou kontrolovanou adresu, která má název IPcheck v address listu se provede „disable“. V opačném případě se provede příkazem „else“ opak, kdy je adresa ve stavu „2“ tak se nastaví proměnná IPUp s komentářem pro danou kontrolovanou adresu a která má název IPcheck v address listu se provede „enable“ a nastaví IPcheck na hodnotu „0“.

V posledních bodech se nastaví plánovač pro IPcheck, aby se spouštěl v lichý čas v intervalu co 2 minuty. Teď už se jen nastaví spuštění initialscript-u, který se provede a následně se odebere. Takto funguje skript pro vytvoření adres pro kontrolu.

```
#IPbuilder
# skript pro vytvoreni testovacich adres z adres listu
# promene

:local coment
:local counter
:local initialscript
:local ip
:global IPcheckName

# nastaveni nazvu a pocitadla
:set IPcheckName ""
:set counter 0

#vymaz existujici skripty s nazvem IPcheck
:foreach i in=[/system script find] do={
:if ([:find [/system script get $i name] "IPcheck"]=0) do={
   /system script remove $i
  }
}

# vymaz IPcheck ze schleduleru
:foreach i in=[/system scheduler find] do={
:if ([:find [/system scheduler get $i name] "IPcheck"]=0) do={
   /system scheduler remove $i
  }
}

# nasataveni IP adresy  komentare
:foreach i in=[/ip firewall address-list find list=IPcheck] do={
    :set ip [/ip firewall address-list get $i address]
    :set coment [/ip firewall address-list get $i comment]
    :set initialscript ($initialscript.":global
IPcheck".$counter."\r\n")
    :if [/ip firewall address-list get $i disabled] do={:set
initialscript ($initialscript.":set IPcheck".$counter." 2\r\n") }

#zde se vytvori podskripty ktere se budou testovat adresy ulozene v
adres listu s nazvem IPcheckX kde X je neustale zvetsujici se cislo
couter
    /system script add name=("IPcheck".$counter)
policy=write,read,test,sniff,policy \
source=("#Kontrola pro ".$coment.", IP adresy ".$ip."\r\n\r\n:global
IPcheck".$counter."\r\n:global IPDown\r\n:global IPUp\r\n\r\n:if
([/ping ".$ip." count=5] = 0) do={\r\n     :if
(\$IPcheck".$counter." = 1) do={\r\n       :set IPcheck".$counter."
2\r\n       :set IPDown (\$IPDown . \"".$coment.", \")\r\n
```

```
:foreach i in=[/ip firewall address-list find list=IPcheck
comment=".$coment."] do={/ip firewall address-list disable \$i}\r\n
}\r\n     :if (\$IPcheck".$counter." = 0) do={:set
IPcheck".$counter." 1}\r\n   } else={\r\n      :if
(\$IPcheck".$counter." = 2) do={\r\n       :set IPUp (\$IPUp .
\"".$coment.", \") \r\n       :foreach i in=[/ip firewall address-
list find list=IPcheck comment=".$coment."] do={/ip firewall
address-list enable \$i}\r\n     }\r\n     :set IPcheck".$counter."
0\r\n   }\r\n}")

   /system scheduler add name=("IPcheck".$counter) start-time=0:1:0
interval=0:2:0 on-event=("IPcheck".$counter)
    :set IPcheckName ($IPcheckName.",".$coment)
    :set counter ($counter+1)
}
/system script add name=IPcheckInitial
policy=ftp,write,read,test,winbox source=($initialscript)
/system script run IPcheckInitial
/system script remove IPcheckInitial
```

4.2.2 Skript ReportIPcheck

Funkce tohoto skriptu již vyplývá z názvu. Bude tedy posílat zprávy o stavu, ve kterém se nachází IP adresa. Nastavením proměnné email deklaruji elektronickou adresu, na kterou má být změna stavu posílaná. Vycházíme zde ze dvou globálních proměnných, které jsou již použité v předchozím podskriptu, který se vygeneruje ze skriptu IPbuilder. Funkce skriptu spočívá v zjištění, zda je proměnná IPDown nebo IPup prázdná. Pokud se tedy IPDown nerovná prázdné hodnotě, jinak řečeno pokud je v IPDown komentář pro danou kontrolovanou adresu tak pomocí příkazu *„/tool email send*“ odešle zprávu, že adresa není v provozu. Podobné to je v případě, uvádění adresy zpět do provozu. Bude-li v IPUp nějaká hodnota pošle zprávu o znovu spuštění adresy do provozu.

Teď už jen stačí nastavit plánovač se spuštěním skriptu v intervalech po 2 minutách a startovacím časem 00:00:00.

```
#reportIPcheck
# skript pro posilani zprav o stavu

:local email martin.svidr@gmail.com

:global IPDown
:global IPUp

:if ($IPDown!="") do={
    /tool e-mail send from=("mikrotik@vsb.cz") to=$email
subject=("Cas: " . [/system clock get time]) body=("IP adresa neni
aktivni:".$IPDown)
   :set IPDown ""
}

:if ($IPUp!="") do={
    /tool e-mail send from=("mikrotik@vsb.cz") to=$email
subject=("Cas: " . [/system clock get time]) body=(" IP adresa je
znovu aktivni:".$IPUp)
   :set IPUp ""
}

/system scheduler add comment="" disabled=no interval=2m
name=reportIPcheck on-event=reportIPcheck start-date=jan/01/1970
start-time=00:00:00
```

4.3 Zálohovací skript

Tento skript patří v mikrotiku k těm jednodušším, ale přesto jeden z těch důležitějších a neobejde se bez něj kterýkoliv Internetový poskytovatel, který používá tyto zařízení.

Skript obsahuje lokální proměnné sysname, time a date, které získávají z potřebných nástrojů v mikrotiku informace. Sysname bere ze „/system identity“ název zařízení, a ze „/system clock se vezmou proměnné date a time. Příkazem „/export file“ vytvoříme soubor s názvem záloha-(sysname). Aby nedošlo k polovičnímu nahrání zálohy a následné její odeslání, je tento problém ošetřen zpožděními jak při nahrávání, tak po něm než bude soubor opět odstraněn. Po vyexportování se provede příkaz „/tool e-mail“, kde je nastaveno od koho byl email odeslán, komu má přijít, nastavení smtp serveru , předmět zprávy, tělo zprávy, ve kterém je zadán datum a čas, a který soubor má být vyexportován. V našem případě tedy zaloha-(sysname). Po dokončení se soubor po prodlevě smaže.[8]

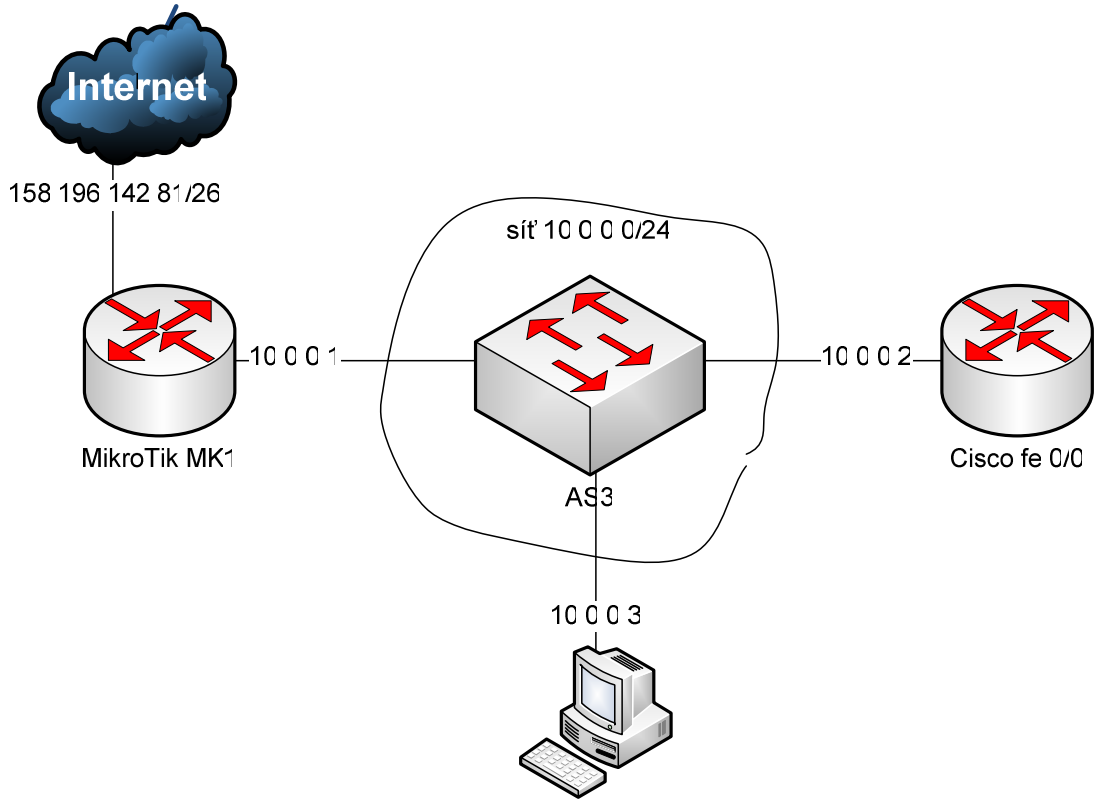
```
#zaloha na mail

# deklarace promennych
:local sysname
:local time
:local date

# nastaveni casu, datumu, systemove jmena
:set sysname [/system identity get name]
:set time [/system clock get time]
:set date [/system clock get date]
/export file=("zaloha-" . $sysname)
#spozdeni pro uplne nahrani zalohy
:delay 15
/tool e-mail send from=($sysname . "@vsb.cz")
to=martin.svidr@gmail.com server=158.196.1.26 \
subject=("Mikrotik " . $sysname . " Zaloha") body=("Zaloha ze dne "
. $date . " " . $time) file=("zaloha-" . $sysname . ".rsc")
:delay 15
file remove ("zaloha-" . $sysname . ".rsc")
#end
```

5. Testování skriptů v reálných podmínkách

V této části mé bakalářské práce, se zabývám otestováním vytvořených skriptů jak pro skript řešící řízení QoS, bezpečnosti i zálohy. Abychom mohli dané skripty otestovat je třeba nejprve na Mikrotiku nastavit, aby vše správně fungovalo. Testování bylo prováděno v laboratoři počítačových sítí na vysoké škole báňské.

5.1 Topologie sítě

Pro nastavení sítě je nejprve vhodné rozvrhnout si správnou topologii sítě, kterou máme na Obr 5.1. Nám bude jako vstupní brána k Internetu sloužit Mikrotik MK1, který bude napojen na switch AS3, který byl umístěn v racku spolu s cisco routerem. Na switch byl také připojen počítač pro monitorování, co prošlo z Internetu přes Mikrotik, a jestli skripty pracují správně.

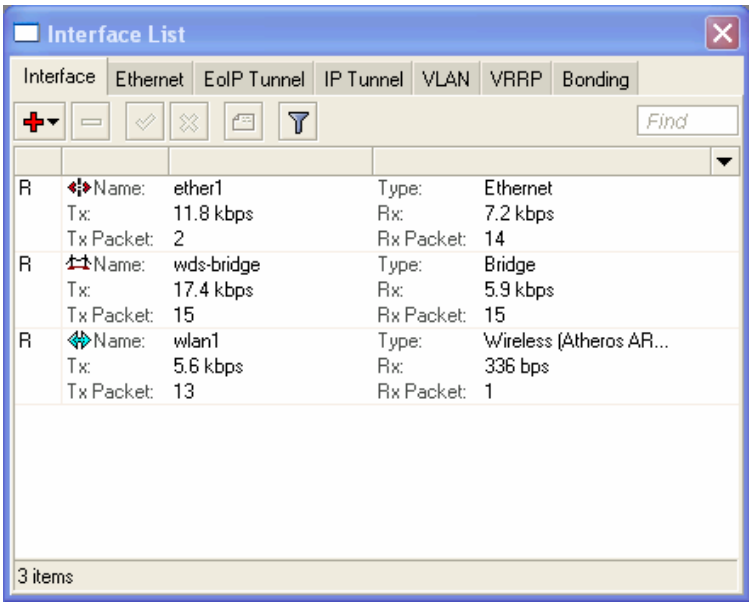


Obr 5.1 Zvolená topologie

5.2 Nastavení Mikrotiku RouteruOS

5.2.1 Nastavení interfaces

Nastavení bylo prováděno v prostředí winbox. Nejprve je třeba nastavit *interfaces* kde jsou všechny hardwarové adaptéry viz Obr 5.2, ale mohou zde být i virtuální. Pro testování naší topologie jsme měli na portu ether4 připojení na switch AS3 a na portu ether5 připojení k Internetu. [10]



Obr 5.2 Interface list v Mikrotiku

Příkazy pro nastavení Mikrotiku pro porty ether4 a ether5:

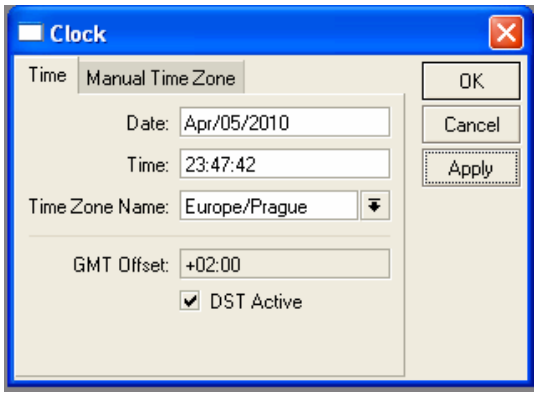
```
add address=158.196.142.81/26 broadcast=158.196.142.81.255
comment="default" disabled=no
interface=ether5
```

add address=10.0.0.1/24 broadcast=10.0.0.255 comment="default" disabled=no interface=ether2 network=10.0.0.0

Nastavení Cisca: interface FastEthernet0/0 ip address 1.0.0.2 255.255.255.0

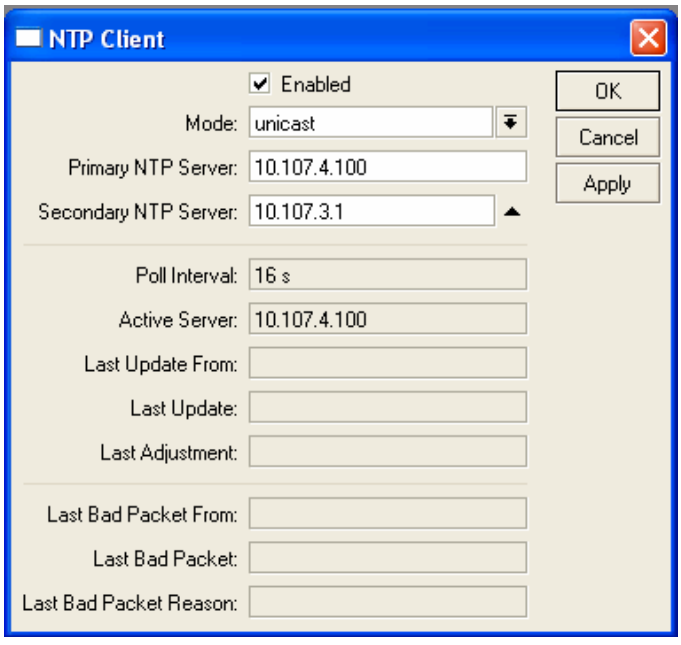
5.2.2 Nastavení hodin

Pro skripty, které byly vytvořeny je důležité nastavení času, jelikož při odesílání zpráv na e-mail, je v některých skriptech použit čas, který se bere z tohoto nastavení. Nastavení hodin je umístěno v menu *System -> Clock*, jež můžeme vidět na Obr 5.3.



Obr 5.3 Nastavení času

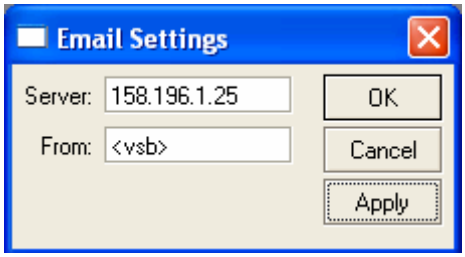
Při vypnutí Mikrotiku ze sítě se pokaždé čas nastaví na tovární nastavení a ukazuje tak na špatný čas. Abychom zamezili tomuto jevu, musíme nastavit NTP klienta na Obr 5.4, který nastaví čas automaticky ze serveru na Internetu, a je rovněž uložen v menu */System* hned pod nastavením času.



Obr 5.4 Nastavení NTP klienta

5.2.3 Nastavení e-mailu

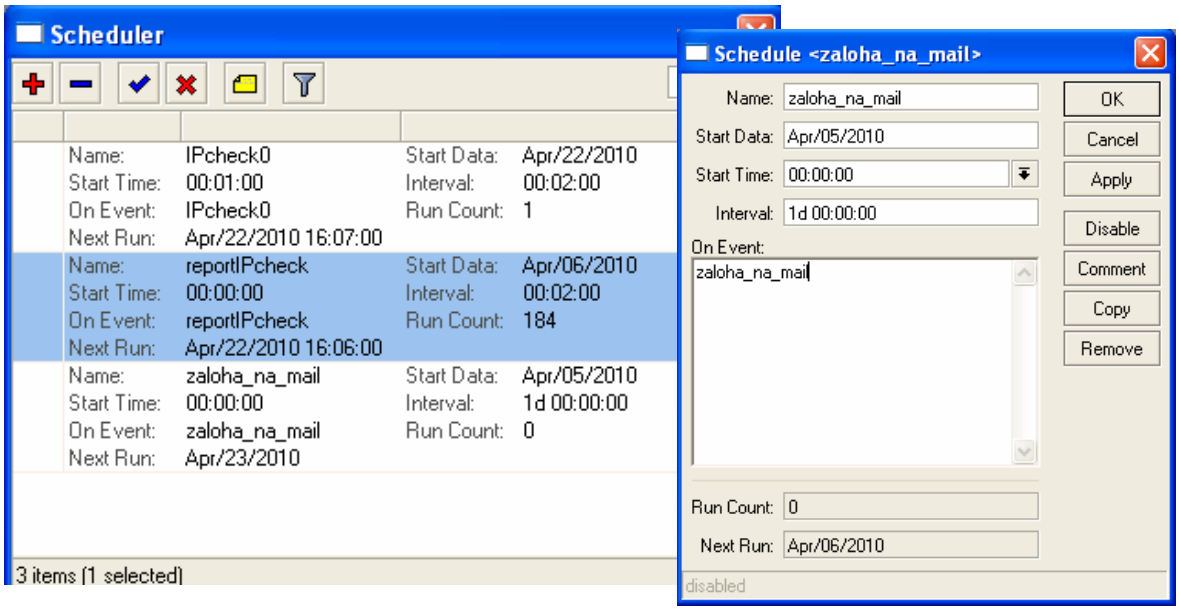
Abych mohl odesílat e-maily, je třeba nastavit SMTP server od vašeho poskytovatele. V našem případě je to SMTP server vysoké školy báňské která má IP adresu 158.196.1.25. K nastavení se dostaneme přes menu v položce *Tools* -> *Email*. Nastavení můžeme vidět na obrázku Obr 5.5.



Obr5.5 Nastavení SMTP serveru

5.2.4 Nastavení Plánovače

Když potřebuje něco naplánovat ,aby se skripty spouštěly tak jak potřebuji je buď na mne, zda udělám skripty s implementovaným plánovačem nebo se jen nastaví ručně. Pro nastavení skriptu ručně se dostaneme přes menu v *System* -> *Schleduler* viz Obr 5.6 . Pro přidání nové úlohy je třeba nastavit Name, Start Data, Start Time Interval a při jaké události se bude plánovač spouštět.

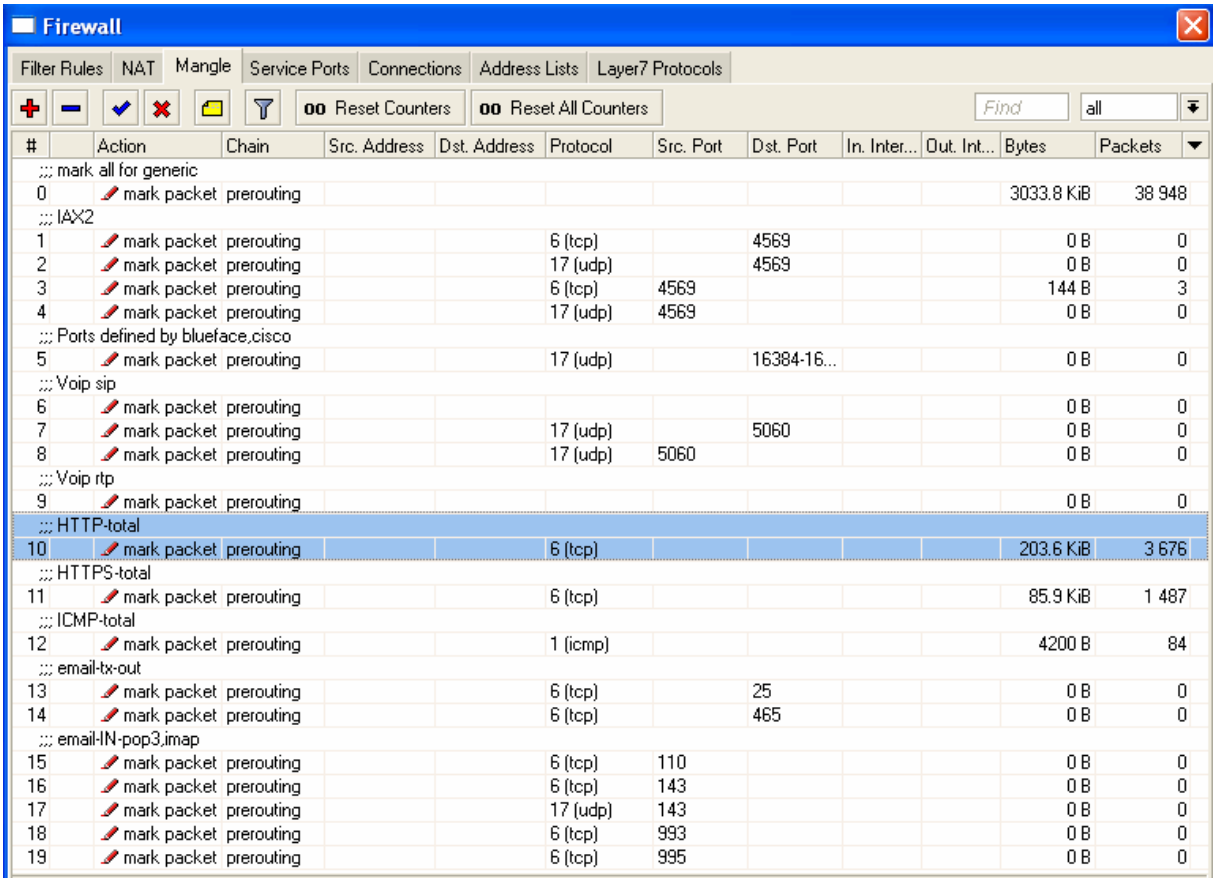


Obr 5.6 Nastavení plánovače

5.3 Testováni skriptu pro QoS

Pro testování tohoto skriptu jsem využil dvou zdrojů, nejprve jsem testoval skript v laboratoři počítačových sítí, kde je ovšem velice rychlý Internet, tudíž omezení ostatních uživatelů zde moc nefungovalo. Rozhodl jsem se tedy otestovat zařízení v klasické domácí síti, kde byl IP telefon. Zapojení vypadalo stejně jako na obrázku topologie sítě s rozdílem, že místo cisco routeru byl umístěn IP telefon.

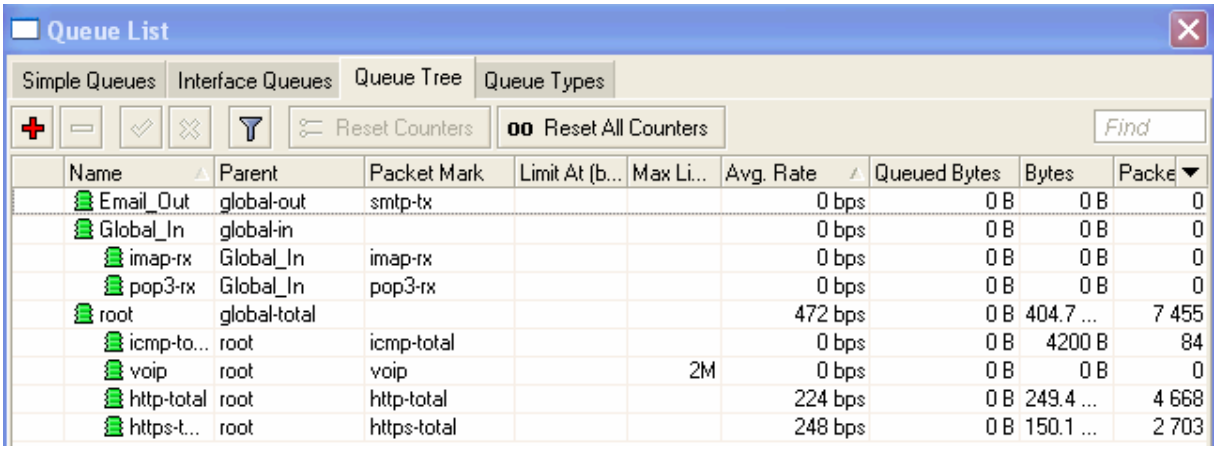
Nejprve jsme spustili skript, který jsem pojmenoval „*rizeni_qos*“ a nechal jej pracovat, ať jde vidět, že se pakety manglují, viz Obr 5.7. Pakety, které zrovna mají zobrazenou 0, nejsou využívány, ale při testování byly všechny odzkoušeny a bezproblémově fungují.[15],[19]



#	Action	Chain	Src. Address	Dst. Address	Protocol	Src. Port	Dst. Port	In. Inter...	Out. Int...	Bytes	Packets
;;; mark all for generic											
0	mark packet	prerouting								3033.8 KiB	38 948
;;; IAX2											
1	mark packet	prerouting			6 (tcp)		4569			0 B	0
2	mark packet	prerouting			17 (udp)		4569			0 B	0
3	mark packet	prerouting			6 (tcp)	4569				144 B	3
4	mark packet	prerouting			17 (udp)	4569				0 B	0
;;; Ports defined by blueface,cisco											
5	mark packet	prerouting			17 (udp)		16384-16...			0 B	0
;;; Voip sip											
6	mark packet	prerouting								0 B	0
7	mark packet	prerouting			17 (udp)		5060			0 B	0
8	mark packet	prerouting			17 (udp)	5060				0 B	0
;;; Voip rtp											
9	mark packet	prerouting								0 B	0
;;; HTTP-total											
10	mark packet	prerouting			6 (tcp)					203.6 KiB	3 676
;;; HTTPS-total											
11	mark packet	prerouting			6 (tcp)					85.9 KiB	1 487
;;; ICMP-total											
12	mark packet	prerouting			1 (icmp)					4200 B	84
;;; email-tx-out											
13	mark packet	prerouting			6 (tcp)		25			0 B	0
14	mark packet	prerouting			6 (tcp)		465			0 B	0
;;; email-IN-pop3,imap											
15	mark packet	prerouting			6 (tcp)	110				0 B	0
16	mark packet	prerouting			6 (tcp)	143				0 B	0
17	mark packet	prerouting			17 (udp)	143				0 B	0
18	mark packet	prerouting			6 (tcp)	993				0 B	0
19	mark packet	prerouting			6 (tcp)	995				0 B	0

Obr 5.7 Manglované pakety v */ip firewall mangle*

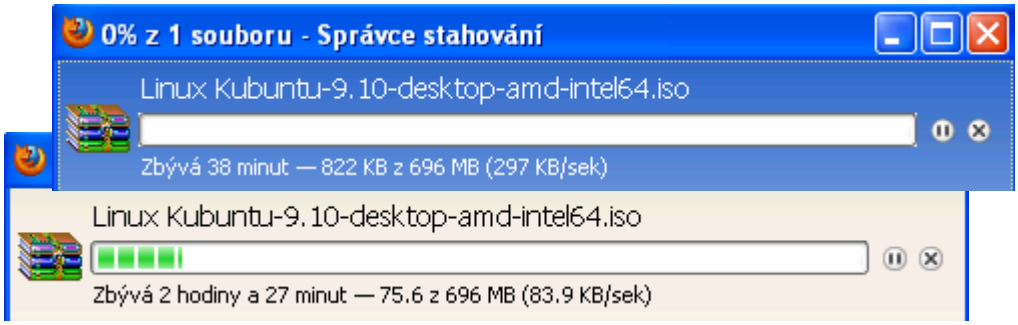
Následně byla vytvořena stromová fronta-QT, kterou si můžeme prohlednout na obrázku Obr 5.8, kde jsou tyto pakety upřednostňovány.



Name	Parent	Packet Mark	Limit At (b...	Max Li...	Avg. Rate	Queued Bytes	Bytes	Packe
Email_Out	global-out	smtp-tx			0 bps	0 B	0 B	0
Global_In	global-in				0 bps	0 B	0 B	0
imap-rx	Global_In	imap-rx			0 bps	0 B	0 B	0
pop3-rx	Global_In	pop3-rx			0 bps	0 B	0 B	0
root	global-total				472 bps	0 B	404.7 ...	7 455
icmp-to...	root	icmp-total			0 bps	0 B	4200 B	84
voip	root	voip		2M	0 bps	0 B	0 B	0
http-total	root	http-total			224 bps	0 B	249.4 ...	4 668
https-t...	root	https-total			248 bps	0 B	150.1 ...	2 703

Obr 5.8 Stromové fronty

Při testování VoIP jsem začal stahovat soubor o velikosti 696mb, který běžel rychlostí 297kbps, když sem začal telefonovat stahování souboru se okamžitě zpomalilo na 83,9kbps, viz Obr5.9. A při hovoru neprobíhalo žádné sekání hovoru, vynechávání písmen, syntetický znějící hlas ani zpoždění.

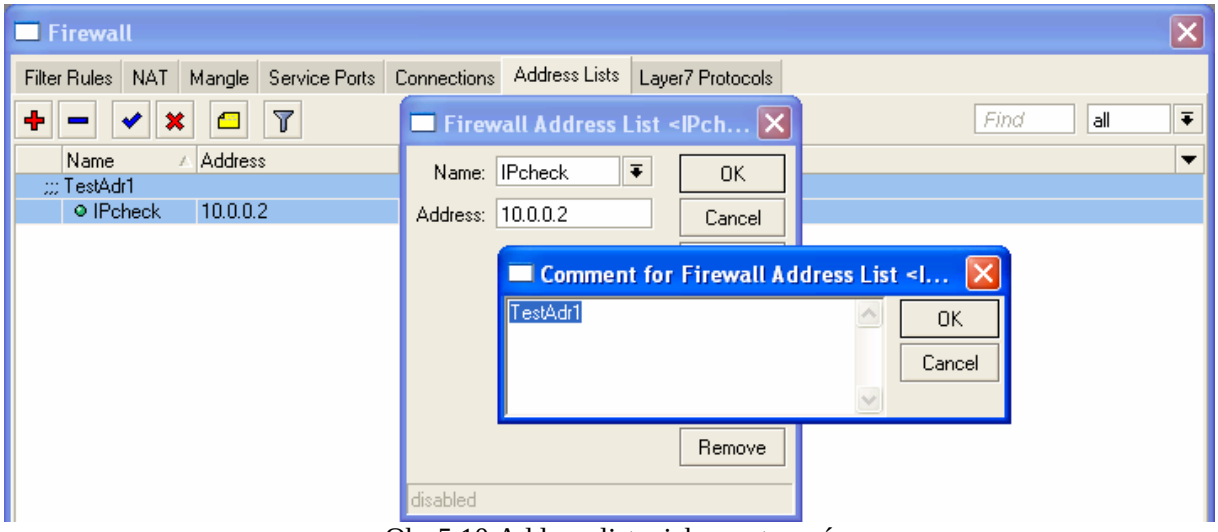


Obr 5.9 Omezení před a po začátku telefonování.

Při tomto testování byl prováděn i ping na zařízení. A to před aplikací QoS a po jeho aplikaci. Při stahování jsem měl vysoké odezvy, po aplikování jsme měl odezvu velice dobrou. Pro jeho jednoduchost jsem grafické zobrazení neuváděl. QoS je potřebná funkce pro všechny uživatele co využívají Hlasové služby, konferenční hovory apod. bez nich by služby pro běžného domácího uživatele s klasickým připojením nepoužitelné.

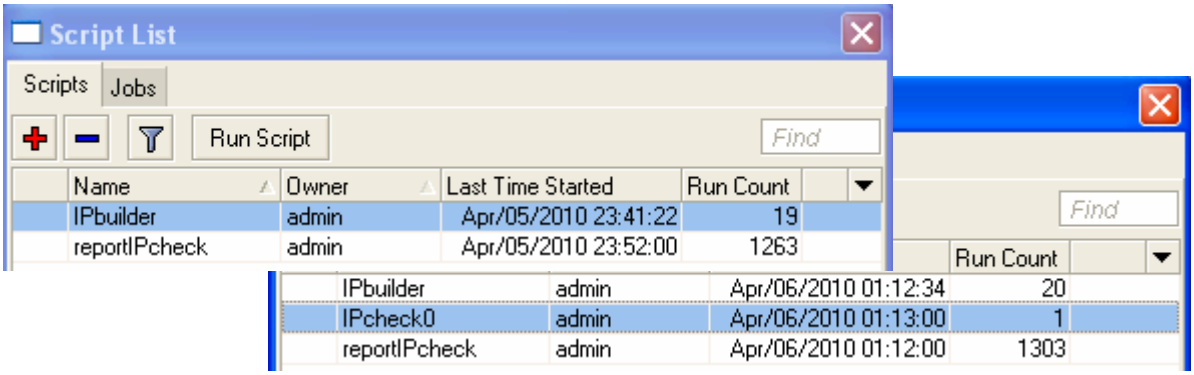
5.4 Testování Bezpečnostního skriptu

Pro testovaní jsem nejprve nahrál skripty na testovaný Mikrotik MK1 (dle topologie). Nahrané skripty byly 2. IPbuilder a reportIPcheck. Nejprve jsem zvolil adresu kterou jsem kontroloval. To sem provedl v menu *IP -> Firewall* v záložce address list, jak můžeme pozorovat na Obr 5.10. Naše testovací adresa je adresa cisco routeru, který má nastaven IP adresu 10.0.0.2 Název vždycky musí být IPcheck pokud chceme aby byl kontrolován.



Obr 5.10 Address list a jeho nastavení

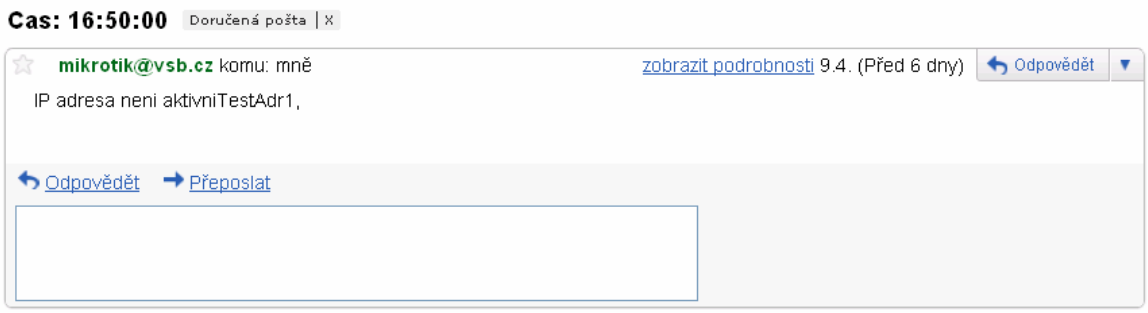
Po nastavení kontrolních adres v address listu můžeme spustit skript IPbuilder, který spustí skripty pro jednotlivé kontrolní Adresy. Jelikož máme v address listu jedinou adresu spustí se tedy pouze jeden skript pod názvem IPcheck0, viz Obr 5.11.



Obr.5.11 Skript IPbuilder před a po spuštění

Skripty mají v sobě implementovaný plánovač tudíž v něm nemusíme nic nastavovat. Pokud se však chceme podívat na nastavení můžeme tak učinit dle „Nastavení Plánovače“ v předchozí kapitole. A jelikož e-mail již máme nastaven všechna nastavení v Mikrotiku jsou pro tento skript nastavena.

Při testování jsem pozoroval, jak skript pracuje a nevyskytly se žádné problémy. Při odpojení síťového LAN kabelu na cca 2minuty se nic nestalo, a jelikož zpráva o zasílání musí projít dvojitou kontrolou a intervaly jsou nastaveny na 2minuty odesílání. Tedy až po delší prodlevě, když jsem LAN kabel odpojil na delší dobu cca 5minut, tak se na e-mail zaslala zpráva o nedostupnosti dané IP adresy,viz Obr 5.12. A adresa se v address listu přepne do stavu „disable“ aby nám nechodily stále emaily o té samé nedostupnosti.



Obr 5.12 email o nedostupném stavu

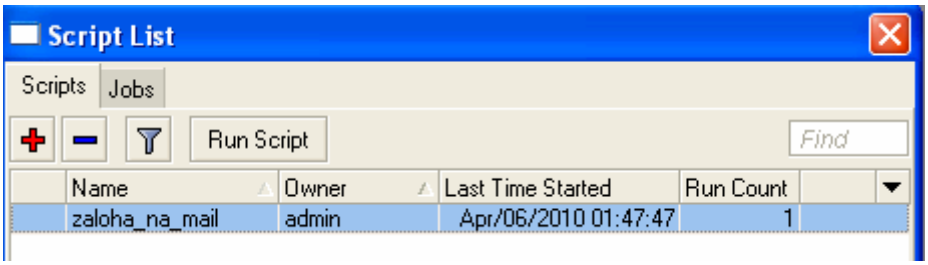
Po znovu zapojení do sítě, vše proběhlo tak jak mělo a bylo mi posláno další upozornění, že adresa je opět aktivní, viz Obr 5.13.

Obr 5.13 email o dostupném stavu

V doručeném emailu se bere komentář, který zadáváme v address listu. Z toho důvodu je vhodné komentářům zadávat přímo nějaké specifické ID, ať víme které zařízení je nedostupné. Pro mé testovací účely postačil komentář TestAdr1.

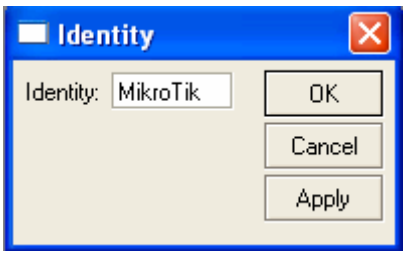
5.5 Testování skriptu pro zálohu

Při testování tohoto skriptu jsem opět nahrál skript na testovací Mikrotik MK1-Obr 5.14. Při prvním testování jsem jej spustil ručně pak jsem již využil služeb plánovače, Obr 5.6, který je nastaven, že se bude spouštět v 1.denním intervalu. Skripty se nahrávají v menu *Systém -> Scripts* Nastavení plánovače je pospáno v předchozích kapitolách.



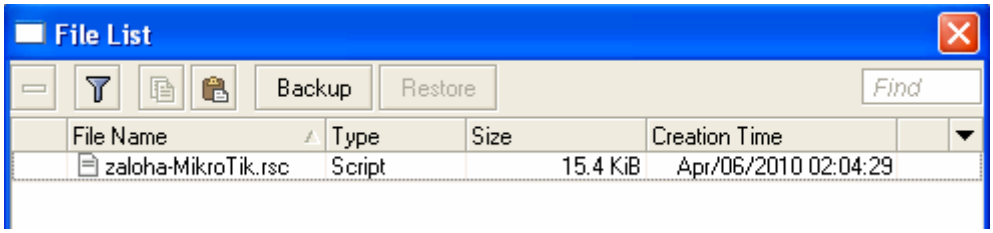
Obr 5.14 Nahrání skriptu.

Před spuštěním ještě musíme nastavit v Mikrotiku název zařízení. Jelikož s touto proměnou se ve skriptu pracuje.Proto tedy v menu *System-> Identity* nastavíme název. Pro nás to bude název „*MikroTik*“, který máme na Obr 5.15.



Obr 5.15 Název Mikrotiku

Nyní můžeme soubor spustit. V menu *Files*, Obr5.16, se vytvoří soubor, který se po odeslání na mail automaticky smaže.



Obr 5.16 File list

Po zkontrolování emailu naleznu poštu se zálohou, viz Obr 5.17.

Obr 5.17 email o úspěšném odeslání zálohy

Závěr

Cílem mé práce bylo seznámení se s platformou Mikrotik RouterOS, s jeho vlastnostmi a s jeho základními i pokročilými funkcemi. Základní funkce jsem použil pro nastavení Mikrotiku tak, aby byl správně nakonfigurován a aby bylo možno funkce později využít v praktické části. K pokročilým funkcím patří skriptování. Skriptování byla v mé práci vyčleněna celá kapitola. První dvě kapitoly jsou tedy teoretické. Mojí snahou bylo co nejefektivněji seznámit uživatele se zařízením Mikrotik RouterOS a se skriptovacím jazykem. Některé funkce, které jsem si odzkoušel, nebyly v praktické části využity. Přesto jsem je popsal v teoretické části, a to z důvodu inspirace pro uživatele, jež by mohli tyto znalosti využít. Práce tudíž může sloužit jako návod nebo inspirace pro začínající správce sítí.

V práci jsem se především zabýval skriptováním v Mikrotiku. Skriptování vyžaduje kompletní znalost syntaxe, korektnost psaní, znalost technické angličtiny a trpělivost při testování. To probíhá takřka „po řádcích“, jelikož při laďění (debugování) programu nepoznáme chybu. Z tohoto důvodu je lepší psát jednotlivé skripty postupně a po každém kroku jej otestovat. Při mém zpracovávání jsem se nechával inspirovat již existujícími skripty a podle nich jsem se učil. Získané znalosti jsem pak použil ve vlastní implementaci. Většina skriptů si jsou podobné a spousty příkazů se tak dá snadno naučit.

V praktické části jsem vytvořil skripty řešící řízení QoS s preferováním VoIP paketů s implementovaným omezením uživatelů. Dále bezpečnostní skript, který kontroluje, zda jsou IP adresy a Routery dostupné. Dále skript řešící zálohu platformy Mikrotik RouterOS a její následné poslání na e-mail. Moje skripty pro zálohu a bezpečnost našly využití v praxi u mého začínajícího kolegy, který používá zařízení Mikrotik. Skript pro QoS také najde uplatnění u uživatelů, jež využívají služby VoIP, nebo u poskytovatelů, kteří limitují své zákazníky podle počtu stažených dat.

V mé práci jsem využíval Mikrotik RB411 a zařízení z laboratoře počítačových sítí. Při konfiguraci jsem používal domácí nastavení, nebo nastavení z laboratoře. Mikrotik se mi po celou dobu jevil jako stabilní zařízení pro mou jednoduchou síť. Všechny skripty byly řádně otestovány.

Mikrotik jako zařízení pro bezdrátové malé a střední sítě se mi jeví jako více než dobré zařízení. Je totiž velice kvalitní. S jeho možnostmi nastavení a s jeho pořizovací cenou vytváří více než ideální řešení. Proto bych Mikrotik doporučil všem malým i středním firmám nebo domácnostem. Tam bych viděl v jeho využití velký potenciál.

Seznam použité literatury:

[1] SLÁVIK, Beno. *Využití speciálního operačního systému „MikroTik RouterOS“ v sítích lokálních ISP*. České Budějovice, 2007. 55 s. Diplomová práce. Jihočeská univerzita.

[2] DVOŘÁČEK, Radek . *Internetový firewall založený na filtrování paketů*. Zlín, 2007. 75 s. Bakalářská práce. Univerzita Tomáše Bati.

[3] *MikroTik web* [online]. 2009 [cit. 2010-03-20]. What is RouterOS. Dostupné z WWW: <http://www.mikrotik.com/pdf/what_is_routeros.pdf>

[4] *MikrTik web* [online]. 2010 [cit. 2010-04-20]. RouterOS features. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Manual:RouterOS_features>.

[5] *MikroTik web* [online]. 2005 [cit. 2010-04-20]. Scripting Host. Dostupné z WWW: <http://www.mikrotik.com/testdocs/ros/2.9/system/scripting.php>.

[6] *MikroTik web* [online]. 2010 [cit. 2010-04-20]. Manual Netwatch. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Netwatch>.

[7] Winbox In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2010-04-29]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Winbox>.

[8] *ISPforum* [online]. 2005 [cit. 2010-04-20]. Záloha na E-mail. Dostupné z WWW: <http://ispforum.cz/viewtopic.php?f=3&t=15>.

[9] ŠÁDEK, Jiří. *Eldar* [online]. 2005 [cit. 2010-03-12]. Quality of Service. Dostupné z WWW: <http://eldar.cz/manasek/felbox/36mps/qos/index.htm>.

[10] *HKfree* [online]. 2007 [cit. 2010-04-11]. RouterBoard. Dostupné z WWW: <http://wiki.hkfree.org/Routerboard>.

[11] *Asm* [online]. 2008 [cit. 2010-04-20]. Mikrotik řízení datových toku. Dostupné z WWW: <http://download.asm.cz/inshop/prod/xtendlan/Mikrotik/EM-Mikrotik-Rizeni_datovych_toku.pdf>.

[12] *ISPforum* [online]. 2005 [cit. 2010-02-15]. Sledování online stavu na e-mail. Dostupné z WWW: <http://ispforum.cz/viewtopic.php?f=3&t=67&st=0&sk=t&sd=a&hilit=online+stavu>.

[13] Session Initiation Protocol In *Wikipedia : the free encyclopedia* [online].Wikipedia Foundation, , [cit. 2010-03-05]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Session_Initiation_Protocol>.

[14] RTP Control Protocol In *Wikipedia : the free encyclopedia* [online]. : Wikipedia Foundation, , [cit. 2010-03-05]. Dostupné z WWW: <http://en.wikipedia.org/wiki/RTP_Control_Protocol>.

[15] List of TCP and UDP port numbers In *Wikipedia : the free encyclopedia* [online].: Wikipedia Foundation. DostupnézWWW: <http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers>

[16] PODGORNÝ, Radek. *Root* [online]. 2003 [cit. 2010-01-18]. HTB-jemný úvod. Dostupné z WWW: <http://www.root.cz/clanky/htb-jemny-uvod/>.

[17] PUŽMANOVÁ, Rita. *TCP/IP v kostce*. [s.l.] : Kopp, 2009. 619 s. ISBN 978-80-7232-388-3.

[18] *MikroTik web* [online]. 2005 [cit. 2010-02-15]. Mangle. Dostupné z WWW: <http://www.mikrotik.com/testdocs/ros/2.9/ip/mangle.php?permalink=0.7790341126868532>.

[19] *Forum MikroTik* [online]. 2009 [cit. 2010-02-15]. Firewall Mangle and Packet Marks problem. Dostupné z :<http://forum.mikrotik.com/viewtopic.php?f=2&t=37080&hilit=drop+firewall+mangle>.

[20] MikroTik RouterOS In *Wikipedia : the free encyclopedia* [online]. Wikipedia Foundation, , [cit. 2010-04-29]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/MikroTik_RouterOS>.

[21] ŠTRAUCH, Adam. *Root* [online]. 2008 [cit. 2010-04-20]. Mikrotik: seznámení s Wi-Fi krabičkou. Dostupné z WWW: <http://www.root.cz/clanky/mikrotik-seznameni-s-wi-fi-krabickou/>.

[22] Wi-fi In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2010-04-29]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Wi-fi>.

[23] MikroTik web [online]. 2009 [cit. 2010-04-29]. Manual Scripting. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Scripting>.

[24] QoS In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2010-04-29]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/QoS>.

Seznam obrázků:

Seznam Příloh

Příloha P1: Skript na řízení QoS s preferováním VoIP

```
#Script na priorizaci komunikacnich kanalu pres IP a omezeni
ostatnich uzivatelu
# Vypracoval Martin Svidrnoch
# zde nastav pri kolika bytech dojde k omezeni

:local limit 2000000000

# zde nastav jaka bude omezena rychlost

:local speed 1000000

# zde nastav na jakou parent se bude omezovani vztahovat

:local parent global-total



# Pracovni promenne
:local par ""
:local traf ""
:local name ""
:local edata ""
:local data ""
:local max ""


#jestlize uz sou mangle obsazeny podle oznacenych paketu uz zadane
nepridavej znova !
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=voip]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=generic]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=http-total]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=https-total]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=icmp-total]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=smtp-tx]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=pop3-rx]
/ip firewall mangle remove [/ip firewall mangle find new-packet-
mark=imap-rx]
```

```
:if ( [/ip firewall mangle find new-packet-mark] = "" ) do={

/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=generic passthrough=yes comment="mark all for generic"
disabled=no
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=voip passthrough=yes dst-port=4569 protocol=tcp \
   comment=IAX2 disabled=no
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=voip passthrough=yes dst-port=4569 protocol=udp
disabled=no
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=voip passthrough=yes src-port=4569 protocol=tcp
disabled=no
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=voip passthrough=yes src-port=4569 protocol=udp
disabled=no

/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=voip passthrough=yes dst-port=16384-16482 protocol=udp \
   comment="Ports defined by blueface,cisco" disabled=no
/ip firewall mangle add chain=prerouting dscp=26 action=mark-packet
new-packet-mark=voip passthrough=yes comment="Voip sip" disabled=no
/ip firewall mangle add chain=prerouting  action=mark-packet new-
packet-mark=voip passthrough=yes dst-port=5060 protocol=udp
disabled=no
/ip firewall mangle add chain=prerouting  action=mark-packet new-
packet-mark=voip passthrough=yes src-port=5060 protocol=udp
disabled=no
/ip firewall mangle add chain=prerouting dscp=46 action=mark-packet
new-packet-mark=voip passthrough=yes comment="Voip rtp" disabled=no
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=http-total passthrough=yes comment="HTTP-total"
disabled=no protocol=tcp port=80
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=https-total passthrough=yes comment="HTTPS-total"
disabled=no protocol=tcp port=443
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=icmp-total passthrough=yes comment="ICMP-total"
disabled=no protocol=icmp
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=smtp-tx passthrough=yes comment="email-tx-out"
disabled=no protocol=tcp dst-port=25
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=smtp-tx passthrough=yes disabled=no protocol=tcp dst-
port=465
```

```
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=pop3-rx passthrough=yes comment="email-IN-pop3,imap"
disabled=no protocol=tcp src-port=110
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=imap-rx passthrough=yes  disabled=no protocol=tcp src-
port=143
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=imap-rx passthrough=yes  disabled=no protocol=udp src-
port=143
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=imap-rx passthrough=yes  disabled=no protocol=tcp src-
port=993
/ip firewall mangle add chain=prerouting action=mark-packet new-
packet-mark=pop3-rx passthrough=yes  disabled=no protocol=tcp src-
port=995
/system scheduler add comment="" name=rizeni_qos disabled=no
interval=5d  on-event=rizeni_qos start-date=jan/01/1970 start-
time=00:00:00
}
#QT
/queue tree add name="root" parent=global-total packet-mark=""
limit-at=0 queue=wireless-default priority=8 max-limit=0 \
   burst-limit=0 burst-threshold=0 burst-time=0s disabled=no

/queue tree add name="icmp-total" parent=root packet-mark="icmp-
total" limit-at=0 queue=wireless-default priority=1 max-limit=0 \
   burst-limit=0 burst-threshold=0 burst-time=0s disabled=no

/queue tree add name="http-total" parent=root packet-mark="http-
total" limit-at=0 queue=wireless-default priority=3 max-limit=0 \
   burst-limit=0 burst-threshold=0 burst-time=0s disabled=no

/queue tree add name="https-total" parent=root packet-mark="https-
total" limit-at=0 queue=wireless-default priority=3 max-limit=0 \
   burst-limit=0 burst-threshold=0 burst-time=0s disabled=no

/queue tree add name="Email_Out" parent=global-out packet-mark=smtp-
tx limit-at=0 queue=default priority=1 max-limit=0 burst-limit=0
burst-threshold=0 burst-time=0s

/queue tree add name="Global_In" parent=global-in limit-at=0
priority=8 max-limit=0 burst-limit=0 burst-threshold=0 burst-time=0s

/queue tree add name="pop3-rx" parent=Global_In packet-mark=pop3-rx
limit-at=0 queue=default priority=1 max-limit=0 burst-limit=0 burst-
threshold=0 burst-time=0s

/queue tree add name="imap-rx" parent=Global_In packet-mark=imap-rx
limit-at=0 queue=default priority=1 max-limit=0 burst-limit=0 burst-
threshold=0 burst-time=0s
```

```
# Vytvoření Queue Tree pro voip
:local unikatnijmeno "voip"
:local priorita "2"



:if ( [/queue tree find packet-mark=([$unikatnijmeno])] = "" ) do={
/queue tree
add name=([$unikatnijmeno]) parent=root packet-
mark=([$unikatnijmeno]) queue=wireless-default priority=$priorita
max-limit=2000000\
burst-limit=0 burst-threshold=0 burst-time=0s disabled=no
} else={
:log error ("Tato queue " . ([$unikatnijmeno]) . " je již použita -
změň prosím unikátní jméno!!")
}

#najdi vsechny položky v QT krom uniatnijemno(voip) a  omez je podle
zadanych promenych
:foreach i in=([/queue tree find packet-mark=!([$unikatnijmeno])] =
"" ) do={

:set name [get $i name]
:set traf [get $i bytes]
:set par [get $i parent]
:set max [get $i max-limit]

:if (($par=$parent) && ($traf>$limit) && !($max = $speed))  do={
   /queue tree set $name max-limit=$speed

   }
}
```